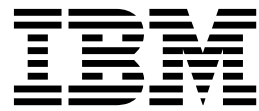Version 11 Release 2

IBM DB2 Administration Tool for z/OS User's Guide

IBM

Version 11 Release 2

IBM DB2 Administration Tool for z/OS User's Guide

IBM

> **Note:**
> Before using this information and the product it supports, read the "Notices" topic at the end of this information.

Contents

About this information

This information provides instructions for customizing and using IBM® DB2®
Administration Tool for z/OS®, a DB2 catalog administration tool.

These topics are designed to help database administrators, system programmers,
and application programmers perform these tasks:

- Plan for the installation of DB2 Admin.
- Install and operate DB2 Admin.
- Customize your DB2 Admin environment.
- Administrate IBM DB2 by using DB2 Admin
- Diagnose and recover from DB2 Admin problems.

Users of this information should understand basic DB2 concepts and facilities.

Always check the DB2 Tools Product publications page for the most current
version of this publication:

http://www.ibm.com/software/data/db2imstools/db2tools-library.html

Chapter 1. DB2 Admin overview

DB2 Admin is a DB2 administration product that can greatly increase the productivity of the entire DB2 staff (database administrators, system administrators, and application developers).

DB2 Admin uses dynamic SQL to access the DB2 catalog tables and to present the information in an easy-to-use ISPF interface.

DB2 Admin is one of several IBM tools that can help you manage database administration and the change management process.

Topics:
- "What's new in DB2 Admin"
- "What does DB2 Admin do?" on page 2
- "Database administration and change management solutions" on page 5
- "DB2 Admin features and benefits" on page 5
- Service updates and support information
- Product documentation and updates
- Accessibility features

What's new in DB2 Admin

This topic summarizes the technical changes for this edition.

New and changed information is indicated by a vertical bar (|) to the left of a change. Editorial changes that have no technical significance are not noted.

Version 11, December 2014, SC19-4134-04

- During customization, you might need to specify the technique for unicode translation. See Required in some cases: Specify a unicode translation technique parameter value for more information.
- Steps on how to migrate from one release or mode of DB2 to another are described in Optional: Migrate modes.

Version 11, August 2014, SC19-4134-03

- Tools Customizer field labels are shortened to increase in the length of the input field. You can see the changed field names in Worksheets: Gathering parameter values for Tools Customizer.
- Control how data is loaded into the target system by using the migrate LOAD utility option REPLACE and RESUME. The steps are described in "Step 2. Generate batch jobs" on page 340.
- Several new Change Management batch interface parameter definitions have been added, for example parameters needed for REORG INDEX. See "Parameter definitions: Change Management batch interface" on page 604 for more information.

Version 11, April 2014, SC19-4134-02

- The step you take to change a foreign key by using the ALTER command has changed and it is shown in "Changing foreign keys" on page 330.

| • Updated information on the types of values that the &CHGTAG. variable resolves to is described in "Product-defined symbol variables: Change Management batch interface" on page 708.
| • Several new Change Management batch interface parameter definitions have been added. See "Parameter definitions: Change Management batch interface" on page 604 for more information.

Version 11, November 2013, SC19-4134-01

• Updated information on filtering objects returned by the LISTDEF based on the format of the RBA or LRSN is described in "Editing a single LISTDEF clause" on page 268.
• Information about viewing and altering the information for group buffer pools. See "Viewing group buffer pools" on page 507 and "Altering group buffer pools" on page 508 for more information.

Version 11, October 2013, SC19-4134-00

• Information about multi-target changes that allow you to register a change to any catalog object on one system and import the change on multiple target systems is described in "Multi-target changes" on page 585.
• Information about using IBM DB2 Analytics Accelerator is described in "Using IBM DB2 Analytics Accelerator" on page 456.
• A new example of changing a unique key and an updated example of changing a foreign key are included in "Examples of redefining a table" on page 309.
• New information about global variables is described in "Displaying global variables and their authorizations" on page 492.
• Updated List Plan Table panel (ADB2EL) in "Listing rows from a plan table" on page 431.
• Updated Explain panel (ADB2E) in "Using the main EXPLAIN panel" on page 429.
• A new example of redefining an index that is defined with the Exclude Null Keys option is included in "Example of redefining an index: Excluding null keys" on page 326.
• A new example of redefining an index or a partitioning index is included in "Redefining an index or a partitioning index" on page 322.
• An updated example of renaming an index is shown in "Renaming an index" on page 318.
• An updated Generate SQL from DB2 Catalog panel (ADB2GEN) is shown in "Generating SQL to re-create a DB2 object" on page 215.
• An updated example of Creating an index without option Exclude/Include null keys is shown in "Creating an index on a table" on page 199.
• An updated scrollable fields on the ALTER Table panel is shown in "Using scrollable fields on DB2 Admin panels" on page 139.

What does DB2 Admin do?

DB2 Admin helps you with the day-to-day tasks associated with managing DB2 environments efficiently and effectively.

DB2 Admin simplifies the complex tasks that are associated with safely managing DB2 objects and schema throughout the application lifecycle with the least possible impact to availability. The key attributes of DB2 Admin include the following:

• Enables quick and easy navigation through the DB2 catalog

- Builds and executes dynamic SQL statements without requiring you to know the exact SQL syntax
- Manages and tracks changes that are made to DB2 object definitions, resolving any potential conflicts before execution
- Helps build DB2 commands to execute against databases and tables
- Builds and executes utility jobs, enabling use of LISTDEFs and TEMPLATEs for increased productivity
- Enables you to create, alter, migrate, drop and reverse engineer DB2 objects

For further details, see the following sections in this topic.

The easy-to-use comprehensive features of DB2 Admin can increase your productivity and increase the reliability of your DB2 objects:

Object management
- Provides in-depth DB2 catalog navigation, which can minimize the time that is required to review the catalog. Objects in the catalog are shown and interpreted, and relevant catalog information is presented logically. You can issue any DB2 command, including BIND, REBIND, and FREE, against selected plans and packages.

  DB2 Admin presents the DB2 catalog quickly and logically:
  - Displays any object in the catalog
  - Displays related DB2 objects by using special line commands
  - Interprets catalog information
  - Displays the authorization for objects
  - Displays the static SQL statements from application plans and packages
  - Displays the DDL for existing views
  - Runs on one of multiple copies of the DB2 system catalog
- Integrated with DB2 utilities to simplify the creation of DB2 utility jobs. JCL can be generated for DB2 utilities and can be executed. The use of LISTDEFs and TEMPLATEs is also supported.
- Enables tasks such as alter, create, drop and migrate of DB2 objects
- Allows reverse engineering of DB2 objects
- Supports DB2 predictive governing
- Enables you to alter the definition of a DB2 table
- Enables you to request the Prompt function, which prompts you before a statement is executed

Security management
- Displays authorizations that have been granted on any type of DB2 object, and enables you to REVOKE these authorizations or GRANT new authorizations
- Provides REVOKE impact analysis to prevent inadvertent data loss when you revoke authorities
- Displays the list of secondary authorization IDs and manages SQL IDs

Performance management
- Allows complex performance and space queries
- Contains a built-in EXPLAIN function that allows you to EXPLAIN a query, and provides an interpretation of the PLAN_TABLE output into an easy-to-understand format
- Comes with a set of performance health check catalog queries

- Enables you to perform space-related functions such as resizing page sets, lets you move page sets to and from STOGROUP- and VCAT-defined space, and helps you estimate space allocations for new table spaces and indexes
- Enables you to create and manage work statement lists (WSLs) and run the WSL as a batch job
- Enables you to dynamically manage system parameters

Change management

- Allows you to manage and track changes to DB2 objects
- Allows you to register changes to multiple target environments
- Allows groups of users to collaborate to build changes by managing information through a series of DB2 tables
- Provides a convenient audit trail that can be used to determine the status of objects that are being changed and where those changes were deployed
- Allows you to recover changes and restore database objects to their previous state

System management

- Allows you to display and cancel threads; display and alter buffer pools; display, start, and stop DB2 traces; and set and display the logs
- Performs various system administration functions, such as updating RLIMITs and managing DDF tables
- Provides a convenient way to administer RLF and DDF tables
- Manages stored procedure operations, such as creating, displaying or altering stored procedures, issuing the DB2 START and STOP STORED PROCEDURE command, and showing statistics for stored procedures that are accessed by DB2 applications
- Displays current dynamic DSNZPARMs change parameters, generates new DSNZPARM modules with changes, and activates those changes in DB2

Application management

- Builds and executes dynamic SQL statements without requiring you to know the exact SQL syntax
- Runs most DB2 utilities
- Enables you to extend existing DB2 Admin applications or to rapidly develop new applications
- Allows you to work with a copy of the DB2 catalog to avoid contention and other performance problems on the actual catalog
- Accesses a remote DB2 catalog where a DDF connection exists between systems. This feature enables you to centrally manage all of your DB2 subsystems with a single DB2 Admin session.
- Allows you to execute any dynamic SQL statement through DB2 Admin, or to invoke SPUFI

Database administration and change management solutions

IBM solutions help IT organizations maximize their investment in DB2 and IMS™ databases and address some of today's toughest IT challenges. Database administration and change management are the core responsibilities of the DBA. If not managed correctly, database administration and change management can monopolize data center resources, waste valuable time, and result in the generation of unwanted errors.

In managing critical database assets and the change management process, DBAs are faced with many challenges. Some examples are as follows:

- Being able to quickly and easily navigate the DB2 catalog
- Ensuring that all of the necessary steps are completed when making a change
- Managing and tracking the changes to the definitions of database objects
- Propagating changes to other database environments
- Keeping DB2 software versions current
- Managing a corrupt database

Many DB2 Tools products provide database management features that are not available in DB2 itself or that provide enhancements to capabilities that are built into DB2.

For example, DB2 Admin allows you to navigate the DB2 catalog quickly and easily.

DB2 Admin provides integration with other DB2 Tools products to create extra function with product-specific line commands for editing tables, analyzing the cost of SQL statements, and analyzing potential access path changes. DB2 Admin offers a central, ISPF-based access point for other DB2 Tools products, such as DB2 Table Editor, DB2 SQL Performance Analyzer, and DB2 High Performance Unload.

DB2 Admin is only one of several DB2 Tools products that provide enhancements to the process of database administration and change management for your databases.

The following DB2 Tools products that can assist with database administration and change management:

- DB2 Object Comparison Tool
- DB2 Storage Management Utility
- Optim™ Test Data Management
- DB2 Table Editor
- DB2 SQL Performance Analyzer
- DB2 High Performance Unload

DB2 Admin features and benefits

The features of DB2 Admin help you to efficiently and effectively manage DB2 environments.

Related concepts:

"What does DB2 Admin do?" on page 2
DB2 Admin helps you with the day-to-day tasks associated with managing DB2 environments efficiently and effectively.

DB2 Admin features

Display the DB2 catalog tables

DB2 Admin provides extensive support for displaying the DB2 catalog. The scope of information that can be displayed is described in this information.

Display any object in the DB2 catalog
> You can retrieve catalog data for any DB2 data object. You can specify the data that is retrieved (for example, you might request that data be retrieved for all databases that are owned by THOMAS and that have the prefix D402).
>
> DB2 Admin retrieves catalog data by using predefined SELECT statements for the more commonly used queries. The rows that are retrieved from the catalog are displayed using the ISPF table-display service. The display panel can be the DB2 Admin default panel, from which you can issue various DB2 Admin line commands, or a panel that you tailor for the result of a particular SQL SELECT. In the latter case, you can use line commands to issue new SQL calls that use information from the columns of rows that have been returned.

Display related DB2 objects using line commands
> You can use DB2 Admin line commands to navigate the catalog. For example, from a display panel that shows databases, you can use a line command to show all table spaces in one of the databases. Then, from the table spaces panel, you could issue a line command to show authorizations for a table space or show the status of image copies for the table space.

Display catalog information
> You can request detailed information about any object in the DB2 catalog. A request for details about an application plan, for example, returns information such as the plan's owner, latest bind time, and number of bytes in the base section.

Show the authorization for DB2 objects
> You can retrieve information about the authorizations for all DB2 objects. From an authorization display panel, you can then grant and revoke privileges.

Display the static SQL from application plans and packages
> You can display the static SQL statements in a plan or a package, which is useful if you do not have access to a program's source code.

Display the DDL for existing views
> You can display the SQL source that created a view, which is useful if you do not have access to the CREATE VIEW SQL (DDL) statement.

Run with multiple copies of the DB2 catalog
> This function allows you to use the DB2 system catalog, one of the many copies of it, or the catalog of a remote site. You might choose to use a different copy of the catalog for each weekday, thus associating a backup with each weekday. Or this feature can allow the system administrator to work on the actual system catalog, while developers use a copy of the catalog, thereby decreasing contention for the catalog.

Execute dynamic SQL statements

You can issue any dynamic SQL statement from your screen or from a data set. You can build and execute an SQL SELECT statement interactively by using line commands.

In addition, by entering required parameters from a panel, you can execute the following SQL statements: GRANT, REVOKE, CREATE, DROP, LABEL ON, and COMMENT ON. This feature allows you to execute the statements without knowing the exact SQL syntax; DB2 Admin provides guidance for the required SQL parameters.

Manage changes to DB2 objects

Use the Change Management function to manage and track the changes that you make to the definitions of your DB2 objects. You can use the Change Management function to complete all of the steps that are typically involved with changing database objects:

1. Defining your changes.
2. Resolving any conflicts by applying any pending changes for the objects as virtual changes.
3. Registering the changes.
4. Analyzing the changes to generate a work statement list that applies the changes.
5. Running the changes in the correct order.

Change Management also makes it easy to back out completed changes. Making and managing changes with Change Management provides a convenient audit trail.

Multi-target change enhances change management and provides the following capabilities:

- Changes can be deployed from one central system to multiple target locations.
- Status and other information about the target change can be communicated back to the central system.
- From one centralized view, DBAs can view of all the changes that have been imported across various target systems.

Issue DB2 commands against databases and table spaces

You can issue any DB2 command against any database or table space that you have selected using DB2 Admin. For example, you can issue the DISPLAY, START, and STOP commands against a database.

DB2 commands are passed to the instrumentation facility interface (IFI), and the result is displayed in ISPF browse.

Run DB2 utilities

You can generate the JCL for DB2 utilities and then run them in batch, or you can include the utility statements in a work statement list to be run at another time or on another subsystem. This function applies to the utilities for storage groups, table spaces, tables, and indexes. For example, you can, generate JCL to run the COPY, REORG, and RUNSTATS utilities for a table space.

The generated JCL consists of a JOB statement, EXEC statement, and all required DD statements. When the JCL is generated, DB2 Admin invokes ISPF edit, which lets you change the JCL, submit it, or copy it to another data set.

You can generate utilities using LISTDEFs and TEMPLATEs.

Issue complex queries

You can run performance and space utilization queries against a database. The data that is returned can help you to determine whether you need to:
- Run the RUNSTATS or STOSPACE utilities
- Reorganize or redesign parts of your database or indexes
- Change the locking rule for tables
- Drop an index
- Move tables to separate table spaces
- Extend the primary allocation for a table space or index
- Reduce the size of a table space

Use the EXPLAIN function

The DB2 Admin EXPLAIN function supports the EXPLAIN statement and provides related support. (The EXPLAIN statement gathers information about the access path DB2 chose to process a query.) By using the EXPLAIN function you can:
- Create a plan table (PLAN_TABLE) in the wanted database and table space.
- Issue an SQL EXPLAIN statement and see the resulting rows in the plan table.
- List a plan table to look at rows from previously executed EXPLAIN statements, or rows from BIND and REBIND operations that were executed with EXPLAIN(YES) specified.

  With this function, predefined search criteria help you find rows in the plan table. Predefined search criteria exist for application plans, DBRMs, collections, and packages. You can see the access path that is chosen by DB2 to process queries, and the tables and indexes that are accessed by DB2.
- Use EXPLAIN (ONLY) to populate EXPLAIN tables but not create a package. This option allows EXPLAIN to be run when the authorization ID of the bind or rebind process does not have the privilege to execute statements in the package.
- Upgrade a plan table to the current version of DB2.
- Look at the DB2 calculated cost.
- Create and display the DB2 explain tables.
- Insert and work with DB2 optimizer hints in the plan table.

Manage SQL IDs

You can change the current DB2 SQL ID by entering a new one or by selecting one from a list of secondary SQL IDs. DB2 Admin displays a list of SQL IDs that you are allowed to use. The list is created either by simulating or invoking the authorization exit in your system.

Perform system administration functions

The system administration functions that you can perform using DB2 Admin include:
- Displaying threads
- Displaying and terminating utilities
- Displaying and managing traces

- Displaying and updating RLIMITs, including the predictive governing limits in DB2
- Displaying and altering buffer pools
- Displaying and setting archive log parameters and archiving the log
- Displaying DB2 system parameters and updating dynamic parameters
- For DDF (distributed data facility):
  - Starting and stopping DDF
  - Displaying and updating the communications database (CDB)
  - Displaying and canceling distributed threads
  - Displaying active locations
- Dynamically managing system parameters

Reverse engineer DB2 objects

Reverse engineering generates the SQL statements necessary to re-create a DB2 object. You can reverse engineer the DB2 objects in your database catalog.

Typical uses for the DB2 Admin reverse engineering function include the following tasks:
- Extracting the DDL for an object before changes are made, so that the changes are applied to the current definition and are available for fallback purposes.
- Moving DB2 objects to another DB2 subsystem. By using the reverse engineering function (together with the table unload and load functions), objects can be moved after a few manual modifications to the generated SQL and batch jobs.

The SQL statements can be generated online or with a batch job.

Use the DB2 predictive governing

You can use DB2 Admin to display, insert, update, or delete predictive governing rows in the resource limit tables. Furthermore, if DB2 Admin receives a predictive governing warning (SQLCODE +495) when running a dynamic SQL statement, DB2 Admin ask whether the SQL statement should be executed or cancelled. If the predictive governing estimates that executing a dynamic SQL statement that was issued from DB2 Admin will exceed the error limit (SQLCODE -495), DB2 Admin displays an error message, and the SQL statement is not executed.

You can use predictive governing limits to prevent users from running *wild* queries on catalog tables or any other tables that aredisplayed using DB2 Admin. By using predictive governing limits for the DB2 Admin package, this type of query can be inhibited either by setting up a predictive governing warning or an error limit in the resource limit table.

Related Reading: For more information on predictive governing, refer to the *DB2 UDB for z/OS Administration Guide*.

Alter the DB2 table definition

You can alter the definition of a DB2 table. Permissible changes include the following tasks:
- Changing the database, table space, owner, and the name of the table
- Modifying the definitions of table columns
- Changing the sequence of the columns in the table
- Inserting and dropping columns

Migrate DB2 data to other DB2 systems

You can copy DB2 data to another DB2 system. This is a useful function if you want to create a separate DB2 test system or if you want to move a test system into production. You can also use this function to consolidate two separate database systems into one.

Extend existing DB2 Admin applications or develop new applications

You can extend DB2 Admin to invoke other ISPF applications that you use for DB2 administration and application development. Some applications that you might want to invoke from DB2 Admin are as follows:
- Security tools
- Vendor DB2 utilities
- Storage management tools

DB2 Admin also enables you to quickly build new ISPF applications for displaying and maintaining DB2 data. Some of the types of data for which you might build such applications are as follows:
- Application definition data
- DB2 performance data
- Extra security data

A sample application is included with the product to illustrate how you might use DB2 Admin to create new applications.

Perform space management functions

DB2 Admin enables you to perform space-related functions such as resizing page sets, moving page sets to and from STOGROUP- and VCAT-defined space, and estimating space allocations for new table spaces and indexes.

Create and run work statement lists

DB2 Admin enables you to create and run work statement lists that include sets of operations. You can execute the entire set, rerun sets, or capture a set of operations that you create on one system for use on another system.

Launch installed IBM DB2 Tools that have an ISPF interface

You can invoke installed IBM DB2 tools that have an ISPF interface—directly from DB2 Admin. The DB2 Admin Launchpad provides a convenient way of creating a centralized ISPF table with the names of your tools. Then, by selecting an entry in this table, you can easily start one of the tools.

Performance

DB2 Admin is equipped with the following performance features:
- DB2 Admin uses dynamic SQL to access the DB2 catalog, which ensures that DB2 always uses the most efficient access path to the catalog (provided RUNSTATS statistics are available for the DB2 optimizer).

- Before DB2 Admin displays information, it does an SQL commit. By doing so, DB2 Admin ensures that a user cannot lock the catalog for long periods of time. If an SQL error occurs, DB2 Admin rolls back the unit of work before it displays any information.
- DB2 Admin has a default limit of 1000 for fetching rows. This limit helps to prevent time-consuming queries. You can change the default of 1000 for an execution of DB2 Admin if more rows are needed. You can set this value permanently or you can set a parameter in the Change DB2 Admin Defaults panel to reset the default value at the next startup.
- You can use DB2 resource limit facilities (RLF) to limit the amount of CPU time that a dynamic SQL statement in DB2 Admin can use - either by using the reactive governor facilities of RLF or by using the predictive governing facilities.
- DB2 Admin can run on a copy of the DB2 catalog. Besides improving performance, running on a copy of the catalog can reduce contention for the catalog. DB2 Admin provides commands to generate jobs to create and populate copies of the DB2 catalog.

Security

DB2 Admin does not expose the security of the DB2 system. All DB2 access is controlled by the existing security provisions of the DB2 system. You can optionally configure DB2 Admin to allow users to execute DDL generated to re-create views that are dropped as a result of dropping other objects. The user can execute this DDL even if they do not have the direct authority. This is done by using *auth-switching* and has its own safeguards to ensure the DDL is not changed from that generated by DB2 Admin. A user must have access to a RACF® (or equivalent) profile to use auth-switching.

DB2 Admin benefits

This section describes a few of the many ways in which DB2 Admin is typically used, and gives examples of specific applications.

Explore databases

DB2 Admin lets you quickly navigate the DB2 catalog and display tables, table columns, and indexes. If you are authorized by DB2, you can also display the content of tables either by doing a simple list of the table or by building SQL statements and executing them against a table.

You can use the DB2 Admin functions to explore unknown databases rapidly or get a quick overview of a database. None of these uses require that you remember the exact syntax of DB2 commands or SQL statements.

Determine and correct problems

You can use DB2 Admin to identify and fix problems with your databases. With its ability to navigate the catalog and use DB2 commands on objects, DB2 Admin can help you discover, analyze, and fix database problems in a more user-friendly fashion than is available with native DB2.

Develop small applications

You can use DB2 Admin to rapidly develop small applications. As you become familiar with the tool, you might find the time that it takes to develop small DB2 Admin dialogs is greatly reduced.

Examples:

- If you have a tool at your installation that manipulates DB2 tables, you can develop your own line command to access it from the DB2 Admin panel that displays tables (implementing the line command as an SMP usermod). Then you can invoke the table tool as a natural follow-on to using DB2 Admin.
- Perhaps you want to generate more DECLARE statements for a PL/I table than is possible with the DB2 DCLGEN tool. You can write an application to invoke DCLGEN directly from the DB2 Admin panel that displays tables. You can also modify the output you receive from DCLGEN to, for example, meet your installation's standards and requirements.
- You might want to build prototypes of SQL SELECT statements. You can build the statements, test them and, when you are satisfied with them, copy the statements to a data set to include in your application program.
- DB2 Admin can help you maintain any DB2 tables that you use for installation standards and special requirements. You can use DB2 Admin to develop a small application that describes all of the applications that you have in the system. Or you can use it to display existing tables that, for example, contain information about DB2 plan performance or batch job execution statistics.

Copy tables from one DB2 system to another

You can use the table utilities that DB2 Admin generates to copy tables from one DB2 system to another. You need to make a few modifications to the generated JCL.

Start DB2 Tools

You can invoke installed IBM DB2 tools that have an ISPF interface directly from DB2 Admin. DB2 Admin guides you through the process of creating a central table with the names of your DB2 utilities. After this table is created, you can select an entry in it to start one of the DB2 tools.

Service updates and support information

Service updates and support information for this product, including software fix packs, PTFs, frequently asked questions (FAQs), technical notes, troubleshooting information, and downloads, are available from the web.

To find service updates and support information, see the following website:

http://www.ibm.com/support/entry/portal/Overview/Software/
Information_Management/DB2_Tools_for_z~OS

Product documentation and updates

DB2 Tools information is available at multiple places on the web. You can receive updates to DB2 Tools information automatically by registering with the IBM My Notifications service.

Information on the web

The DB2 Tools Product Documentation web page provides current product documentation that you can view, print, and download. To locate publications with the most up-to-date information, refer to the following web page:

http://www.ibm.com/software/data/db2imstools/db2tools-library.html

You can also access documentation for many DB2 Tools from IBM Knowledge Center:

http://www.ibm.com/support/knowledgecenter

Search for a specific DB2 Tool product or browse the **Information Management** > **DB2 for z/OS family**.

IBM Redbooks® publications that cover DB2 Tools are available from the following web page:

http://www.redbooks.ibm.com

The Data Management Tools Solutions website shows how IBM solutions can help IT organizations maximize their investment in DB2 databases while staying ahead of today's top data management challenges:

http://www.ibm.com/software/data/db2imstools/solutions/index.html

Receiving documentation updates automatically

To automatically receive emails that notify you when new technote documents are released, when existing product documentation is updated, and when new product documentation is available, you can register with the IBM My Notifications service. You can customize the service so that you receive information about only those IBM products that you specify.

To register with the My Notifications service:
1. Go to http://www.ibm.com/support/mysupport
2. Enter your IBM ID and password, or create one by clicking **register now**.
3. When the My Notifications page is displayed, click **Subscribe** to select those products that you want to receive information updates about. The DB2 Tools option is located under **Software** > **Information Management**.
4. Click **Continue** to specify the types of updates that you want to receive.
5. Click **Submit** to save your profile.

How to send your comments

Your feedback is important in helping to provide the most accurate and high-quality information. If you have any comments about this book or any other IBM product documentation, use one of the following options:
- Use the online reader comment form, which is located at http://www.ibm.com/software/data/rcf/.
- Send your comments by email to comments@us.ibm.com. Include the name of the book, the part number of the book, the version of the product that you are using, and, if applicable, the specific location of the text you are commenting on, for example, a page number or table number.

Accessibility features

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use a software product successfully.

The major accessibility features in this product enable users to perform the following activities:

- Use assistive technologies such as screen readers and screen magnifier software. Consult the assistive technology documentation for specific information when using it to access z/OS interfaces.
- Customize display attributes such as color, contrast, and font size.
- Operate specific or equivalent features by using only the keyboard. Refer to the following publications for information about accessing ISPF interfaces:
  - *z/OS ISPF User's Guide, Volume 1*
  - *z/OS TSO/E Primer*
  - *z/OS TSO/E User's Guide*

  These guides describe how to use the ISPF interface, including the use of keyboard shortcuts or function keys (PF keys), include the default settings for the PF keys, and explain how to modify their functions.

Chapter 2. Preparing to customize DB2 Admin

Before you start to customize DB2 Admin, determine all of the customization values that you need to specify during the customization process, and familiarize yourself with all of the customization tasks.

The following checklist lists and describes each significant customization step. Use this checklist to guide you through the entire customization process.

Tip: Print the following checklist and the data set names and parameter values worksheets. Use the worksheets to record your values, and refer to them during the customization process.

Task	Link to detailed instructions	Status
Tools Customizer basics		
Prior to beginning the customization process, familiarize yourself with Tools Customizer terminology and data sets, and other basic information about Tools Customizer.	"Tools Customizer terminology and data sets" on page 963	
You also might want to watch a video to familiarize yourself with the customization process.	In IBM developerWorks®, in the DB2 for z/OS Best Practices community: Upgrading DB2 Administration Tool V10.2 to V11.1 by using IBM Tools Customizer The videos are also available on YouTube. The videos were created based on a specific PTF level of DB2 Administration Tool V10.2 and DB2 Administration Tool V11.1. Therefore, the panels in the video may not match the panels in the current releases of the products.	
Hardware requirements		
Ensure that you deploy DB2 Admin on a z-series processor that is capable of running z/OS Version 1 Release 12 or higher.	None.	
Software requirements		
Verify that your environment meets the minimum software requirements. To install and use DB2 Admin, your environment must be running a supported version of the z/OS operating system and of DB2 for z/OS. If you will enable DB2 Object Comparison Tool, DB2 Cloning Tool, DB2 High Performance Unload, or DB2 Table Editor to be launched from DB2 Administration Tool, ensure that you are running the following supported versions: • DB2 Object Comparison Tool V11.2 • DB2 Cloning Tool V3.1 • DB2 High Performance Unload V4.2 • DB2 Table Editor V4.4	"Verify that your environment meets software requirements" on page 18	
SMP/E installation		

Task	Link to detailed instructions	Status
Verify that DB2 Admin has been installed correctly. DB2 Admin is installed by using standard SMP/E processing.	"Verify that DB2 Admin has been installed successfully" on page 19	
Verify that Tools Customizer for z/OS has been installed correctly. Tools Customizer for z/OS is installed by using standard SMP/E processing.	"Verify that Tools Customizer has been installed successfully" on page 19	
Upgrading to newer versions or modes of DB2		
Follow the steps in "Optional: Migrate modes" on page 100 to migrate from one release or mode of DB2 to another DB2 release or mode.	"Optional: Migrate modes" on page 100	
Gather data set names		
During the customization process, you must specify names for the following types of data sets: • Tools Customizer • DB2 Admin	"Worksheets: Gathering required data set names" on page 19	
APF authorization		
The SADBLINK data set must be APF authorized. Alternatively, copy the ADB2ATH and ADB2UTIL programs to an APF-authorized library.	None.	
Gather parameter values		
During the customization process, you must specify parameter values for DB2 Admin, for DB2, and for your LPAR.	Worksheets: Gathering parameter values for Tools Customizer	
Optional: Customize products that will be launched from DB2 Admin		
If you will enable DB2 Object Comparison Tool, DB2 Cloning Tool, DB2 High Performance Unload, or DB2 Table Editor to be launched from DB2 Admin, customize these products before you customize DB2 Admin.-	None.	
Customize DB2 Admin		
Complete the steps in the appropriate customization roadmap based on the type of customization that you are performing.		
Customizing DB2 Admin for the first time Follow this roadmap if you do not have a customized version of DB2 Admin, and you need to customize it for the first time.	Roadmap: Customizing DB2 Admin for the first time	
Customizing a different version of DB2 Admin Follow this roadmap if you have already customized a version of DB2 Admin and you want to use the same parameter values to customize a different version.	"Roadmap: Migrating to DB2 Admin V11.2 from a previous release." on page 72	
Recustomizing DB2 Admin Follow this roadmap if you have a customized version of DB2 Admin but you want to change one or more parameter values.	"Roadmap: Recustomizing DB2 Admin 11.2" on page 75	

Task	Link to detailed instructions	Status
Some customization options require you to manually complete additional tasks after you have used Tools Customizer. If you generated jobs in Tools Customizer that correspond to the following customization options, complete the additional tasks before you submit the jobs. In some cases, an optional task can be completed either by using Tools Customizer or by manually completing tasks without using Tools Customizer.		
Required in some cases: Updating the APF Authorization table		
Update SYS1.PARMLIB to authorize the ADB2ATH and ADB2UTIL programs and the ADB2ATH and ADB2UTIL TSO commands.	"Required in some cases: Update the APF Authorization table" on page 98	
Required in some cases: Specify a unicode translation technique parameter value		
Derive the unicode translation technique from the CCSID conversion string.	Required in some cases: Specify a unicode translation technique parameter value	
Optional: Migrate modes		
Migrate from one release or mode of DB2 to another.	Optional: Migrate modes	
Required in some cases: Tailor DB2 Admin Authorization Switching		
DB2 Admin Authorization Switching is a facility within DB2 Admin to run DDL under the authority of another user.	"Required in some cases: Tailor DB2 Admin Authorization Switching" on page 100	
Optional: Prepare ADBL CLIST		
The ADBL CLIST in the SADBCLST library invokes the DB2 Admin main menu.	"Optional: Prepare ADBL CLIST" on page 101	
Optional: Verify activation of limited functionality		
To use DB2 Admin you must check your TSO LOGON PROC and your link list definition to verify that the DB2 libraries are available to your TSO session.	"Optional: Verify activation of limited functionality" on page 107	
Optional: Tailor the DB2 Admin Launchpad		
The DB2 Admin Launchpad enables you to launch all installed IBM DB2 tools that have an ISPF interface directly from a centralized panel.	"Optional: Tailor the DB2 Admin Launchpad" on page 112	
Optional: Grant SELECT access on catalog tables		
If you plan to make DB2 Admin available to a large number of users, you might want to specify those IDs that are authorized to see the catalog. To complete this step, run the Tools Customizer job with the template ADBGC.	"Optional: Grant SELECT access on catalog tables" on page 112	
Optional: Define Reverse Engineering stored procedure for CC/390		
Apply Reverse Engineering to additional software products, such as Control Center OS/390 (CC/390).	Optional: Define Reverse Engineering stored procedure for CC/390	
Optional: Optimize DSNWZP and DSNZPARM settings		
To optimize performance, verify that the DSNWZP stored procedure is operational and that the DSNZPARM STORTIME(DSN6SYSP) parameter is set at a proper level. You might want to lower the value of the DSNZPARM STORTIME(DSN6SYSP) parameter.	"Optional: Optimize DSNWZP and DSNZPARM settings" on page 113	
Optional: Run the RUNSTATS utility		

Task	Link to detailed instructions	Status
It is recommended that you run the RUNSTATS utility on the DB2 catalog to optimize performance.	None.	
Optional: Enabling DB2 Admin distributed support		
By using distributed support and the Change Management functionality, you can register a multi-target change on a target system using DRDA® access.	"Optional: Enabling DB2 Admin distributed support" on page 113	
Optional: Make DB2 Admin available to users		
You can make DB2 Admin available to users.	"Optional: Make DB2 Admin available to users" on page 114	
Optional: Making Object Comparison Tool available from DB2 Administration Tool		
You can make the DB2 Object Comparison Tool available from DB2 Admin as part of the DB2 Admin customization process. You can also customize the Object Comparison Tool separately from the customization of DB2 Admin.	"Optional: Making Object Comparison Tool available from DB2 Administration Tool" on page 114	
Optional: Make the DB2I and Object Comparison Tool available from the DB2 Administration Tool		
You can make the DB2I and Object Comparison Tool available from the main menu of the DB2 Admin Tool.	"Optional: Make the DB2I and Object Comparison Tool available from the DB2 Administration Tool" on page 115	

Set up your environment prior to customization

Prior to beginning the customization process, ensure that your environment meets all requirements, that you have installed all prerequisite software, and that you have considered how you want to customize optional features.

Verify that your environment meets software requirements

Ensure that you are using z/OS V1.12 (5694-A01) or later.

IBM System Modification Program Extended (SMP/E) for z/OS, V3.5 or higher (5655-G44)

Ensure that you are using one of the following supported versions of DB2 for z/OS:
- DB2 V10 (5605-DB2)
- DB2 Value Unit Edition V10.1 (5697-P31)
- DB2 V11 (5615-DB2)
- DB2 Value Unit Edition V11.1 (5697-P43)

To use the Change Management function in IBM DB2 Object Comparison Tool for z/OS, ensure that you are using DB2 Object Comparison Tool for z/OS V11.2 (5655-DOC).

To clone objects by using IBM DB2 Cloning Tool for z/OS, ensure that you are using DB2 Cloning Tool for z/OS V3.1 (5655-N15).

To unload objects by using IBM DB2 High Performance Unload for z/OS, ensure that you are using DB2 High Performance Unload for z/OS V4.2 (5655-AA1).

To quickly access, update, and delete data by using IBM DB2 Table Editor for z/OS, ensure that you are using DB2 Table Editor for z/OS V4.4 (5697-G65).

Verify that DB2 Admin has been installed successfully

See the Program Directory for IBM DB2 IBM DB2 Administration Tool for z/OS for z/OS, GI10-8972 for installation instructions.

Verify that Tools Customizer has been installed successfully

Tools Customizer for z/OS is a component of IBM Tools Base for z/OS (5655-V93), which is available free of charge. Tools Customizer for z/OS provides a standard approach to customizing IBM DB2 for z/OS Tools.

See the Program Directory for IBM Tools Base for z/OS, GI10-8819 for installation instructions.

Worksheets: Gathering required data set names

Identify and record the data set names that will be used during the customization process and make sure that requirements for certain data sets are met.

Data set names for Tools Customizer

Identify and record the following Tools Customizer data set names:

Data set name	Description	Special requirements	Your data set name
SCCQEXEC	EXEC library for Tools Customizer	None.	
SCCQDENU	Metadata library for Tools Customizer	None.	
SCCQLOAD	Executable load module library for Tools Customizer	None.	
SCCQMENU	ISPF messages for Tools Customizer	None.	
SCCQPENU	ISPF panels for Tools Customizer	None.	
SCCQSAMP	Sample members for Tools Customizer	None.	
SCCQTENU	Table library for Tools Customizer		

Worksheets: Gathering parameter values for Tools Customizer

During the customization process, you will need to provide parameter values for the product that you are customizing, for DB2, and for your LPAR.

Use the worksheets in this topic to record the appropriate parameter settings for your purposes, and then use these worksheets during the customization process. The worksheets are organized based on the order of the customization panels in Tools Customizer.

Settings for Tools Customizer

Description

Use the following worksheet to identify and record the values for Tools Customizer settings. During the customization process, you enter these values on the Tools Customizer Settings panel (CCQPSET).

For more information about the parameters in this section, see "Data sets that Tools Customizer uses during customization" on page 965

Product Customization Settings

Parameter	Sample or default value	Your value
Customization library qualifier The high-level qualifier that is used as the prefix for the output data set that is dynamically generated during the customization process.	DB2TOOL.PRODUCT.CUST	
Use DB2 group attach Determines the value that is used in the CONNECT statements in the generated customization jobs.	YES	

Tools Customizer Library Settings

Parameter	Sample or default value	Your value
Metadata library The fully qualified name of the Tools Customizer SCCQDENU data set.	DB2TOOL.CCQ110.SCCQDENU	
Discover output data set The fully qualified name of the data set in which the output from the DB2 Admin Discover EXEC is stored. This data set is dynamically generated during the customization process.	DB2TOOL.CCQ110.DISCOVER	
Data store data set The fully qualified name of the output data set where Tools Customizer stores information about product or component, LPAR, and DB2 parameter values. This data set is dynamically generated during the customization process.	DB2TOOL.CCQ110.DATASTOR	

User Job Card Settings for Customization Jobs

Parameter	Sample or default value	Your value
The job card information to be inserted into the generated jobs for customizing a product or component.	The job statement information from the ISPF Batch Selection panel.	

Metadata library for DB2 Admin

Description
Use the following worksheet to identify and record the value of the metadata library for DB2 Admin. During the customization process, you will enter this value on the Specify the Metadata Library panel (CCQPHLQ).

DB2 Admin metadata library

Parameter	Sample or default value	Your value
Metadata library The fully qualified name of the DB2 Admin SADBDENU data set.	*hlq*.SADBDENU	

Customization values for the Discover EXEC

Complete this worksheet only if you are recustomizing a product that has previously been customized by using Tools Customizer.

Description
Use the following worksheet to identify and record the customization values for the Tools Customizer Discover EXEC. During the customization process, you will enter these values on the Discover Customized Product Information panel (CCQPDSC).

Discover EXEC for Extracting Information from an Already Customized Product parameters

Parameter	Sample or default value	Your value
Discover EXEC library The fully qualified data set name that contains the product Discover EXEC. If the data set name exceeds 42 characters, enclose the name in quotation marks.	*hlq*.SADBEXEC	
Discover output data set The fully qualified name of the data set for the output from the product Discover EXEC. If the data set name exceeds 42 characters, enclose the name in quotation marks.	The name of the Discover output library that you entered on the Tools Customizer Settings (CCQPSET) panel.	

Information for Discover EXEC parameters

Parameter	Sample or default value	Your value
Source Customized table library Enter the fully qualified name of the DB2 Administration Tool table library, generally from a previous release. If the DISCOVER process detects member ADBTPARM, it is read and populates the Tools Customizer input fields accordingly. If member ADBTPARM is not detected and ADB2PARM/ ADB2DB2D are found instead, then an upgrade from V10.1 install is assumed and will populate the Tools Customizer input fields accordingly.	No default	
Target Customized table library Enter the fully qualified name of the DB2 Administration Tool table library where the Customization table member ADBTPARM will be written to.	No default	
DB2 Group Attach Name	NONE	
Trace This option dumps diagnostic information to a temp file that is specific to DISCOVER processing which later can be used by an IBM representative upon request.	No default	

Product parameters

Description

The parameters that are listed in the Product parameters section on the Customizer Workplace panel (CCQPWRK) are read-only. They contain information that was provided on other panels, by Tools Customizer, or by the DB2 Admin metadata data set.

Read-only Product parameters

Parameter	Discovered?	Source of this value
Product metadata library This value is the library that you specified on the Specify the product parameters panel. This field is scrollable. Place your cursor anywhere on the field and press PF11 to view its full contents.	No	This value is specified on the Specify the Metadata Library panel (CCQPHLQ).
LPAR The **LPAR** field displays the LPAR on which you are customizing DB2 Admin.	No	This value is provided by Tools Customizer.
Product name This value displays the product that is being customized. In this example, DB2 Admin should be displayed in this field. This field is scrollable. Place your cursor anywhere on the field and press PF11 to view its full contents.	No	This value is provided by the product metadata file.
Version The **Version** field displays the version, release and maintenance of the product you are customizing in the format V*n*.R*n.nn*.	No	This value is provided by the product metadata file.
Product customization library This value displays the name of the data set in which the generated library customization jobs will be stored.	No	This value is derived from the user-specified customization library qualifier on the Tools Customizer Settings panel (CCQPSET).

Task: General customization

Description
> Customizes the general DB2 Administration Tool parameters.
>
> This task is required.

Jobs generated
> ADBCUS*ab*, where *ab* are alphanumeric characters assigned by Tools Customizer. This job is based on the ADBCUST template and is in the *job_sequence_number*CUST*DB2_entry_ID* member.

Required authority
> The user ID that runs the ADBCUS*ab* job must have SYSADM or equivalent authority.

Steps and parameters for the General customization task

Step or parameter	Required?	Discovered?	Default value	Your value
Admin Tool/OC CLIST Specify the data set name of the DB2 Admin Tool CLIST library (SADBCLST) and the Object Comparison CLIST library (SGOCCLST) to be used by generated job templates.	Yes	No	No	

Steps and parameters for the General customization task

Step or parameter	Required?	Discovered?	Default value	Your value
Admin Tool DBRM Specify the data set name of the DB2 Admin Tool DBRM library (SADBDBRM) to be used by generated job templates.	Yes	No	No	
Admin Tool/OC EXEC Specify the data set name of the DB2 Admin Tool EXEC library (SADBEXEC) and the Object Comparison EXEC library (SGOCEXEC) to be used by generated job templates.	Yes	No	No	
Admin Tool/OC Load Specify the data set name of the DB2 Admin Tool Load library (SADBLLIB) and the Object Comparison Load library (SGOCLLIB) to be used by generated job templates.	Yes	No	No	
Admin Tool/OC Message Specify the data set name of the DB2 Admin Tool Message library (SADBMLIB) and the Object Comparison Message library (SGOCMLIB) to be used by generated job templates.	Yes	No	No	
Admin Tool/OC Panel Specify the data set name of the DB2 Admin Tool Panel library (SADBPLIB) and the Object Comparison Panel library (SGOCPLIB) to be used by generated job templates.	Yes	No	No	
Admin Tool/OC Skeleton Specify the data set name of the DB2 Admin Tool Skeleton library (SADBSLIB) and the Object Comparison Skeleton library (SGOCSLIB) to be used by generated job templates.	Yes	No	No	
Admin Tool/OC Table Specify the data set name of the DB2 Admin Tool Table library (SADBTLIB) and the Object Comparison Table library (SGOCTLIB) to be used by generated job templates.	Yes	No	No	

Steps and parameters for the General customization task

Step or parameter	Required?	Discovered?	Default value	Your value
Customized Table lib Enter the table library that contains the Customized table ADBTPARM. In general, this is the Target Customized table library that's specified during the DISCOVER process. For new installs, the DISCOVER option is not applicable. Therefore, it is recommended to specify the DB2 Administration Tool table library.	Yes	No	No default	
Admin Tool HLQ The high-level qualifier of the DB2 Administration Tool data sets that will be used by the generated customization jobs.	Yes	No	ADBB10	
CM Batch PROCLIB Enter the name of the library that will contain DB2 Admin JCL procedures. If left blank the default is (Admin HLQ).ADBTCZ.PROCLIB.	Yes	Yes	ADBB10.ADBTCZ. PROCLIB	
System identification method The method that is used to ensure that the batch utility jobs that are created by DB2 Admin will run on the same z/OS system as the DB2 subsystem. To ensure that the same system is used, a /*JOBPARM SYSAFF line is added to the JCL. The following values are valid: **SMFID** Uses the SMF ID. This value is valid only if SMF is active. **JESID** Uses the JES2 ID. This value is valid only on JES2 systems. **NONE** Does not include a /*JOBPARM SYSAFF line in the generated JCL. **SYSNAME** Uses the z/OS system name from the CVT control block. **<name>** <name> is the SYSAFF name.	Yes	Yes	JESID	

Steps and parameters for the General customization task

Step or parameter	Required?	Discovered?	Default value	Your value
Type of DB2 security exit The type of DB2 security exit that is installed for the DB2 subsystem. Valid values are: • STD: Standard DB2 security exit (default) • SAMPLE: Sample DB2 security exit (logic being simulated) • AUTH: Local DB2 security exit that must run authorized • NOCALL: Do not call the security exit. DB2 Admin Tool cannot show SQL IDs. • OWN: Local DB2 security exit that can run unauthorized.	Yes	Yes	STD	
Installation name The installation name is a text string that will be carried forward by DB2 Admin and can be used in local modifications.	Yes	No	No default	
JES node name Enter the JES node name of the remote DB2 subsystem (blank if local). Specify the same value that you would specify on either a JES2 /*XMIT or a JES3 //XMIT DEST=nnnn JECL statement.	Yes	Yes	No default	
Utility data set prefix Subsystem default high-level qualifier (HLQ) of the data sets that are used in DB2 utility jobs. Valid values are: USERID, OWNER, CREATEDBY, or name (use name as HLQ).	Yes	No	No default	
Job class for DB2 utilities Default Job class to be used for running DB2 utilities. Enter a valid value of 1 character.	Yes	Yes	A	
SYSAFF for DB2 utilities The SYSAFF job parameter to be used for batch DB2 Utility jobs. This parameter ensures that batch DB2 Utility jobs are run on the same operating system as the DB2 subsystem. Enter a valid value of 1 - 4 characters.	Yes	Yes	No default	

Steps and parameters for the General customization task

Step or parameter	Required?	Discovered?	Default value	Your value
DB2 Admin APF library Used for: (1)Authorization Switching when building ALTER JCL, and (2)Modules ADB2ATH and ADB2UTIL that otherwise should be in the link list.	Yes	Yes	No default	
JES3 system JES3 system identifies if you are running JES3 or not. Specify YES for JES3.	Yes	Yes	NO	
Remote DB2 subsystem name DB2 subsystem name of the remote DB2 subsystem. Leave blank if local.	Yes	Yes	No default	
Remote location name DB2 location name of the remote DB2 subsystem. Leave blank if local. Specify the value that is defined in the LOCATION column of the SYS1.LOCATIONS table in your DB2 catalog.	Yes	Yes	No default	
Authorization switching Specify YES to enable the Authorization Switching function for the current DB2 subsystem. Specify NO to disable Authorization Switching.	Yes	Yes	NO	
ISPF application ID Identifies the member name in which the ISPF profile variables are saved for the DB2 Administration tool. The default value is null with an application ID of ISR. If you use a minus sign with this parameter, the value set for this parameter is overridden by the DB2 Administration Tool, which is ISR.	Yes	Yes	No default	
Value for PROMPT Options The subsystem default value for Prompt Options. Specify YES or No.	Yes	Yes	NO	
Reset to defaults at startup The subsystem default value for the Reset to Default at Startup parameter. Specify YES or No.	Yes	Yes	YES	
Number of DSNUPROC procedure job steps The number of job steps in the DSNUPROC procedure.	Yes	No	No default	

Steps and parameters for the General customization task

Step or parameter	Required?	Discovered?	Default value	Your value
SSID switching Allows switch of SSID for DB2 subsystems. Specify YES or No.	Yes	Yes	YES	
Authorization switching ID DB2 Security ID to use for auth-switching.	Yes	No	No default	
DB2 CONCENTRATE STATEMENTS WITH LITERALS Default DB2 CONCENTRATE STATEMENTS WITH LITERALS attribute on all dynamic SQL statements. The default is YES. Valid only with DB2 V10 or higher.	Yes	Yes	YES	
DB2 use CONCURRENT clause on SQL Default DB2 CONCURRENTLY COMMITTED attribute on all dynamic SQL statements. The default is YES. Valid only with DB2 V10 or higher.	Yes	Yes	YES	
User cmds lib(mbr) User commands library and member.	Yes	-	No default	
Automatic deletion of compare results Enter "YES" if you want to automatically delete saved compare results as part of the DB2 Adminstration Tool's cleansing process.	Yes	Yes	YES	
High Performance Unload (HPU) enabled Subsystem default to use HPU for Unloads.	Yes	Yes	YES	
HPU load library Subsystem default data set name for the High Performance Unload (HPU) SINZLINK load library when HPU is enabled. This variable is ignored if HPU is not enabled. Do not specify the HPU SINZLOAD data set, since this may cause an abend because of APF-authorization issues. You can specify multiple values for this parameter.	Yes	Yes	No default	

Steps and parameters for the General customization task

Step or parameter	Required?	Discovered?	Default value	Your value
HPU parameter library Subsystem default data set name for the High Performance Unload (HPU) SINZPARM parm library when HPU is enabled. This variable is ignored if HPU is not enabled. Do not specify the HPU SINZLOAD data set, since this may cause an abend because of APF-authorization issues. You can specify multiple values for this parameter.	Yes	Yes	No default	
REXX user exit lib The data set names for the REXX user exits used to specify overwrite values for masking fields DSSIZE, PRIQTY, SECQTY, DEFER, and DEFINE. You can specify multiple values for this parameter.	Yes	Yes	No default	
Enable DB2 Cloning Tool Launch DB2 Cloning Tool from within DB2 Administration Tool as an optional choice for migrating objects/data. Select YES to enable this option, NO to disable.	Yes	Yes	NO	
Cloning Tool CLIST lib Specify the CLIST library that contains the DB2 Cloning Tool invocation CLIST.	Yes	Yes	No default	
Enable DB2 Table Editor Launch DB2 Table Editor from within DB2 Administration Tool as an optional choice to quickly access, update, and delete data. Select YES to enable this option, NO to disable.	Yes	Yes	NO	
Table Editor CLIST(mbr) Specify the startup clist used to invoke the DB2 Table Editor. For example: hlvlqual.SETISAMP(ETI)	Yes	Yes	No default	
Option 1 Will produce an additional menu option to display on the DB2 Administration Menu panel ADB2. Enter a valid value of 1 - 72 characters.	Yes	Yes	No default	

Steps and parameters for the General customization task

Step or parameter	Required?	Discovered?	Default value	Your value
Option 1 description A description of the menu option to be displayed on the DB2 Administration Menu panel ADB2. Enter a valid value of 1 - 72 characters.	Yes	Yes	No default	
ISPF statement for option 1 The ISPF statement that the DB2 Administration Tool will execute for this menu option. Enter a valid value of 1 - 72 characters.	Yes	Yes	No default	
SQL statement for option 1 The SQL statement that the DB2 Administration Tool will execute for this menu option. Enter a valid value of 1 - 256 characters.	Yes	Yes	No default	
DB2 Admin Tool command for option 1 The DB2 Administration Tool command for this menu option. Enter a valid value of 1 - 256 characters.	Yes	Yes	No default	
New DB2 attachment for option 1 Specify YES to start a new DB2 attachment for this menu option. Otherwise, specify NO.	Yes	Yes	No default	
Option 2 Will produce an additional option to display on the DB2 Administration Menu panel ADB2. Enter a valid value of 1 - 72 characters.	Yes	Yes	No default	
Option 2 description A description of the menu option to be displayed on the DB2 Administration Menu panel ADB2. Enter a valid value of 1 - 72 characters.	Yes	Yes	No default	
ISPF statement for option 2 The ISPF statement that the DB2 Administration Tool will execute for this menu option. Enter a valid value of 1 - 72 characters.	Yes	Yes	No default	
SQL statement for option 2 The SQL statement that the DB2 Administration Tool will execute for this menu option. Enter a valid value of 1 - 256 characters.	Yes	Yes	No default	

Steps and parameters for the General customization task

Step or parameter	Required?	Discovered?	Default value	Your value
DB2 Admin Tool command for option 2 The DB2 Administration Tool command for this menu option. Enter a valid value of 1 - 256 characters.	Yes	Yes	No default	
New DB2 attachment for option 2 Specify YES to start a new DB2 attachment for this menu option. Otherwise, specify NO.	Yes	Yes	No default	
Option 3 Will produce an additional option to display on the DB2 Administration Menu panel ADB2. Enter a valid value of 1 - 72 characters.	Yes	Yes	No default	
Option 3 description A description of the menu option to be displayed on the DB2 Administration Menu panel ADB2. Enter a valid value of 1 - 72 characters.	Yes	Yes	No default	
ISPF statement for option 3 The ISPF statement that the DB2 Administration Tool will execute for this menu option. Enter a valid value of 1 - 72 characters.	Yes	Yes	No default	
SQL statement for option 3 The SQL statement that the DB2 Administration Tool will execute for this menu option. Enter a valid value of 1 - 256 characters.	Yes	Yes	No default	
DB2 Admin Tool command for option 3 The DB2 Administration Tool command for this menu option. Enter a valid value of 1 - 256 characters.	Yes	Yes	No default	
New DB2 attachment for option 3 Specify YES to start a new DB2 attachment for this menu option. Otherwise, specify NO.	Yes	Yes	No default	
Option 4 Will produce an additional option to display on the DB2 Administration Menu panel ADB2. Enter a valid value of 1 - 72 characters.	Yes	Yes	No default	

Steps and parameters for the General customization task

Step or parameter	Required?	Discovered?	Default value	Your value
Option 4 description A description of the menu option to be displayed on the DB2 Administration Menu panel ADB2. Enter a valid value of 1 - 72 characters.	Yes	Yes	No default	
ISPF statement for option 4 The ISPF statement that the DB2 Administration Tool will execute for this menu option. Enter a valid value of 1 - 72 characters.	Yes	Yes	No default	
SQL statement for option 4 The SQL statement that the DB2 Administration Tool will execute for this menu option. Enter a valid value of 1 - 256 characters.	Yes	Yes	No default	
DB2 Admin Tool command for option 4 The DB2 Administration Tool command for this menu option. Enter a valid value of 1 - 256 characters.	Yes	Yes	No default	
New DB2 attachment for option 4 Specify YES to start a new DB2 attachment for this menu option. Otherwise, specify NO.	Yes	Yes	No default	
Option 5 Will produce an additional option to display on the DB2 Administration Menu panel ADB2. Enter a valid value of 1 - 72 characters.	Yes	Yes	No default	
Option 5 description A description of the menu option to be displayed on the DB2 Administration Menu panel ADB2. Enter a valid value of 1 - 72 characters.	Yes	Yes	No default	
ISPF statement for option 5 The ISPF statement that the DB2 Administration Tool will execute for this menu option. Enter a valid value of 1 - 72 characters.	Yes	Yes	No default	
SQL statement for option 5 The SQL statement that the DB2 Administration Tool will execute for this menu option. Enter a valid value of 1 - 256 characters.	Yes	Yes	No default	

Steps and parameters for the General customization task

Step or parameter	Required?	Discovered?	Default value	Your value
DB2 Admin Tool command for option 5 The DB2 Administration Tool command for this menu option. Enter a valid value of 1 - 256 characters.	Yes	Yes	No default	
New DB2 attachment for option 5 Specify YES to start a new DB2 attachment for this menu option. Otherwise, specify NO.	Yes	Yes	No default	
Option 6 Will produce an additional option to display on the DB2 Administration Menu panel ADB2. Enter a valid value of 1 - 72 characters.	Yes	Yes	No default	
Option 6 description A description of the menu option to be displayed on the DB2 Administration Menu panel ADB2. Enter a valid value of 1 - 72 characters.	Yes	Yes	No default	
ISPF statement for option 6 The ISPF statement that the DB2 Administration Tool will execute for this menu option. Enter a valid value of 1 - 72 characters.	Yes	Yes	No default	
SQL statement for option 6 The SQL statement that the DB2 Administration Tool will execute for this menu option. Enter a valid value of 1 - 256 characters.	Yes	Yes	No default	
DB2 Admin Tool command for option 6 The DB2 Administration Tool command for this menu option. Enter a valid value of 1 - 256 characters.	Yes	Yes	No default	
New DB2 attachment for option 6 Specify YES to start a new DB2 attachment for this menu option. Otherwise, specify NO.	Yes	Yes	No default	
Option 7 Will produce an additional option to display on the DB2 Administration Menu panel ADB2. Enter a valid value of 1 - 72 characters.	Yes	Yes	No default	

Steps and parameters for the General customization task

Step or parameter	Required?	Discovered?	Default value	Your value
Option 7 description A description of the menu option to be displayed on the DB2 Administration Menu panel ADB2. Enter a valid value of 1 - 72 characters.	Yes	Yes	No default	
ISPF statement for option 7 The ISPF statement that the DB2 Administration Tool will execute for this menu option. Enter a valid value of 1 - 72 characters.	Yes	Yes	No default	
SQL statement for option 7 The SQL statement that the DB2 Administration Tool will execute for this menu option. Enter a valid value of 1 - 256 characters.	Yes	Yes	No default	
DB2 Admin Tool command for option 7 The DB2 Administration Tool command for this menu option. Enter a valid value of 1 - 256 characters.	Yes	Yes	No default	
New DB2 attachment for option 7 Specify YES to start a new DB2 attachment for this menu option. Otherwise, specify NO.	Yes	Yes	No default	
Option 8 Will produce an additional option to display on the DB2 Administration Menu panel ADB2. Enter a valid value of 1 - 72 characters.	Yes	Yes	No default	
Option 8 description A description of the menu option to be displayed on the DB2 Administration Menu panel ADB2. Enter a valid value of 1 - 72 characters.	Yes	Yes	No default	
ISPF statement for option 8 The ISPF statement that the DB2 Administration Tool will execute for this menu option. Enter a valid value of 1 - 72 characters.	Yes	Yes	No default	
SQL statement for option 8 The SQL statement that the DB2 Administration Tool will execute for this menu option. Enter a valid value of 1 - 256 characters.	Yes	Yes	No default	

Steps and parameters for the General customization task

Step or parameter	Required?	Discovered?	Default value	Your value
DB2 Admin Tool command for option 8 The DB2 Administration Tool command for this menu option. Enter a valid value of 1 - 256 characters.	Yes	Yes	No default	
New DB2 attachment for option 8 Specify YES to start a new DB2 attachment for this menu option. Otherwise, specify NO.	Yes	Yes	No default	
Option 9 Will produce an additional option to display on the DB2 Administration Menu panel ADB2. Enter a valid value of 1 - 72 characters.	Yes	Yes	No default	
Option 9 description A description of the menu option to be displayed on the DB2 Administration Menu panel ADB2. Enter a valid value of 1 - 72 characters.	Yes	Yes	No default	
ISPF statement for option 9 The ISPF statement that the DB2 Administration Tool will execute for this menu option. Enter a valid value of 1 - 72 characters.	Yes	Yes	No default	
SQL statement for option 9 The SQL statement that the DB2 Administration Tool will execute for this menu option. Enter a valid value of 1 - 256 characters.	Yes	Yes	No default	
DB2 Admin Tool command for option 9 The DB2 Administration Tool command for this menu option. Enter a valid value of 1 - 256 characters.	Yes	Yes	No default	
New DB2 attachment for option 9 Specify YES to start a new DB2 attachment for this menu option. Otherwise, specify NO.	Yes	Yes	No default	
Option 10 Will produce an additional option to display on the DB2 Administration Menu panel ADB2. Enter a valid value of 1 - 72 characters.	Yes	Yes	No default	

Steps and parameters for the General customization task

Step or parameter	Required?	Discovered?	Default value	Your value
Option 10 description A description of the menu option to be displayed on the DB2 Administration Menu panel ADB2. Enter a valid value of 1 - 72 characters.	Yes	Yes	No default	
ISPF statement for option 10 The ISPF statement that the DB2 Administration Tool will execute for this menu option. Enter a valid value of 1 - 72 characters.	Yes	Yes	No default	
SQL statement for option 10 The SQL statement that the DB2 Administration Tool will execute for this menu option. Enter a valid value of 1 - 256 characters.	Yes	Yes	No default	
DB2 Admin Tool command for option 10 The DB2 Administration Tool command for this menu option. Enter a valid value of 1 - 256 characters.	Yes	Yes	No default	
New DB2 attachment for option 10 Specify YES to start a new DB2 attachment for this menu option. Otherwise, specify NO.	Yes	Yes	No default	

Task: Copy Fixed-Blocked (FB) CLIST/EXEC libraries to Variable-Blocked (VB)

Description
> Creates VB versions of the CLIST and EXEC libraries. The data set names of the new VB libraries are the same as the FB libraries, but are suffixed with ".VB".

> This task is optional.

Jobs generated
> ADBFB2VB. This job is based on the ADBFB2VB template and is in the *job_sequence_number*FB2VB member.

Required authority
> None.

Steps and parameters for the Copy Fixed-Blocked (FB) CLIST/EXEC libraries to Variable-Blocked (VB) task

Step or parameter	Required?	Discovered?	Default value	Your value
Customize Copy FB CLIST and EXEC product libraries to VB.	Yes	No	Selected	

Steps and parameters for the Copy Fixed-Blocked (FB) CLIST/EXEC libraries to Variable-Blocked (VB) task

Step or parameter	Required?	Discovered?	Default value	Your value
VOLSER The volume serial number (VOLSER) used for this task. Specifying a value is not required if managed by Storage Management Subsystem (SMS).	Yes	Yes	No default	
DASD unit The DASD unit used for this task. Specifying a value is not required if managed by Storage Management Subsystem (SMS).	Yes	Yes	No default	

Admin Tool Setup Task: Create and Upgrade

Description

Create and Upgrade objects that are used by the DB2 Administration Tool.

This task is required.

Jobs generated

ADBSETUP. This job processes the following parameters:

- Change Management database
- Checkpoint database
- Catalog copy database
- Profiles history database
- Reverse engineering objects
- Stored procedure ADB2RCP
- VIEW RUNSTATS objects
- Indexes
- GRANT on DB2 catalog tables

Required authority

The user ID that runs this job must have SYSADM or equivalent authority.

Parameter: Change Management database

Description

Create and Upgrade Change Management database objects. Use the Change Management database to manage and track the changes that are made to your DB2 objects.

This parameter is optional.

Steps and parameters for specifying the Change Management database

Step or parameter	Required?	Discovered?	Default value	Your value
Change Management database Specify YES to Create and Upgrade the Change Management database. Specify NO to not Create and Upgrade the Change Management database.	Yes	Yes	Yes	

Steps and parameters for specifying the Change Management database

Step or parameter	Required?	Discovered?	Default value	Your value
Owner name Used by SET CURRENT SQLID to set the owner name upon creation of the database objects. Enter a valid value of 1 - 128 characters.	Yes	Yes	ADB	
Database name Name of the database where the objects and data will be stored. Enter a valid value of 1 - 8 characters.	Yes	Yes	ADBDCHG	
STOGROUP name The name of the storage group (STOGROUP) that is used when creating the database objects. Enter a valid value of 1 - 8 characters.	Yes	Yes	ADBGCHG	
STOGROUP volumes Defines the volumes of the STOGROUP that is used when creating the database objects. Enter a list of one or more VOLSERs separated by commas. The maximum input field length is 128 characters.	Yes	Yes	"*"	
STOGROUP VCAT A catalog name that is used to identify the VSAM Catalog (VCAT) for the STOGROUP. Enter a valid value of 1 - 8 characters.	Yes	Yes	DB2	
Tablespace name prefix The table space objects that will be created with a name prefixed with 1 - 4 characters.	Yes	Yes	ADBS	
Tablespace BUFFERPOOL name The buffer pool to be used when creating the table space objects. Valid values are: BP0 - BP49, BP8K0 - BP8K9, BP16K0 - BP16K9, BP32K, and BP32K1 - BP32K9.	Yes	No	No default	
Index BUFFERPOOL name The buffer pool to be used when creating the index objects. Valid values are: BP0 - BP49, BP8K0 - BP8K9, BP16K0 - BP16K9, BP32K, and BP32K1 - BP32K9.	Yes	No	No default	
Enable Change Management Set this parameter to YES if you intend to use Change Management for every DB2 subsystem.	Yes	No	YES	

Steps and parameters for specifying the Change Management database

Step or parameter	Required?	Discovered?	Default value	Your value
Enable Allow Change Delete Enable the delete change line command, but only if the Change Management database objects owner is not blank.	Yes	No	NO	
One PROCLIB for multiple SSIDs Specify whether the CM Batch procedure is to support multiple DB2 subsystems.	Yes	No	YES	
CM Batch JCL procedure name The name of the CM Batch JCL procedure when one procedure will be used to support multiple SSIDs.	Yes	No	GOCCM	
CM Batch local installation parameters The name of the data set that contains your default parameters for CM Batch.	Yes	No	No default	

Parameter: Checkpoint database

Description

> Create and Upgrade the checkpoint database. The information to monitor the execution of the input stream is stored in a DB2 table referred to as the checkpoint table. The DB2 Admin Batch Restart program, ADBTEP2, enables an execution restart or resume of an input stream of SQL statements, utilities, and DB2 commands in a batch job at an intermediate point, if any one of the statements in that input stream fails.

> This parameter is required.

Steps and parameters for specifying the Checkpoint database

Step or parameter	Required?	Discovered?	Default value	Your value
Checkpoint database You must specify YES to Create and Upgrade the Checkpoint database. Note that this is required for DB2 Admin Tool, not optional.	Yes	Yes	Yes	
Owner name Used by SET CURRENT SQLID to set the owner name upon creation of the database objects. Enter a valid value of 1 - 128 characters.	Yes	Yes	ADB	
Database name Name of the database where the objects and data will be stored. Enter a valid value of 1 - 8 characters.	Yes	Yes	ADBDCH	

Steps and parameters for specifying the Checkpoint database

Step or parameter	Required?	Discovered?	Default value	Your value
STOGROUP name The name of the Storage Group (STOGROUP) that will be used when creating the database objects. Enter a valid value of 1 - 8 characters.	Yes	Yes	ADBGCH	
STOGROUP volumes Defines the volumes of the STOGROUP that will be used when creating the database objects. Enter a list of one or more VOLSERs separated by commas. The maximum input field length is 128 characters.	Yes	Yes	"*"	
STOGROUP VCAT A catalog name that is used to identify the VSAM Catalog (VCAT) for the STOGROUP. Enter a valid value of 1 - 8 characters.	Yes	Yes	DB2	
Tablespace name prefix The table space objects that will be created with a name prefixed with 1 - 6 characters.	Yes	Yes	ADBSCH	
Tablespace BUFFERPOOL name The buffer pool to be used when creating the table space objects Valid values are: BP0 - BP49, BP8K0 - BP8K9, BP16K0 - BP16K9, BP32K, and BP32K1 - BP32K9.	Yes	Yes	No default	
Index BUFFERPOOL name The buffer pool to be used when creating the index objects. Valid values are: BP0 - BP49, BP8K0 - BP8K9, BP16K0 - BP16K9, BP32K, and BP32K1 - BP32K9.	Yes	Yes	No default	

Parameter: Catalog Copy database

Description
> Create and Upgrade the Catalog Copy database. This information tracks which DB2 catalog copies are available for use. This parameter also adds a CC option on the DB2 Admin Tool Primary Menu. On the menu, you can display and manage the DB2 Catalog Copy Version table. If this option is disabled, the CC option does not appear on the DB2 Admin Tool Primary Menu.

> This parameter is optional.

Steps and parameters for specifying the Catalog Copy database

Step or parameter	Required?	Discovered?	Default value	Your value
Catalog Copy database Specify YES to Create and Upgrade the Catalog Copy database. Specify NO to not Create and Upgrade the Catalog Copy database.	Yes	Yes	Yes	
Owner name Used by SET CURRENT SQLID to set the owner name upon creation of the database objects. Enter a valid value of 1 - 128 characters.	Yes	No	ADB	
Database name Name of the database where the objects and data will be stored. Enter a valid value of 1 - 8 characters.	Yes	Yes	ADBDCC	
STOGROUP name The name of the Storage Group (STOGROUP) that will be used when creating the database objects. Enter a valid value of 1 - 8 characters.	Yes	Yes	ADBGCC	
STOGROUP volumes Defines the volumes of the STOGROUP that will be used when creating the database objects. Enter a list of one or more VOLSERs separated by commas. The maximum input field length is 128 characters.	Yes	Yes	"*"	
STOGROUP VCAT A catalog name that is used to identify the VSAM Catalog (VCAT) for the STOGROUP. Enter a valid value of 1 - 8 characters.	Yes	Yes	DB2	
Tablespace name prefix The table space objects that will be created with a name prefixed with 1 - 6 characters.	Yes	Yes	ADBSCC	
Tablespace BUFFERPOOL name The buffer pool to be used when creating the table space objects. Valid values are: BP0 - BP49, BP8K0 - BP8K9, BP16K0 - BP16K9, BP32K, and BP32K1 - BP32K9.	Yes	No	No default	

Steps and parameters for specifying the Catalog Copy database

Step or parameter	Required?	Discovered?	Default value	Your value
Index BUFFERPOOL name The buffer pool to be used when creating the index objects. Valid values are: BP0 - BP49, BP8K0 - BP8K9, BP16K0 - BP16K9, BP32K, and BP32K1 - BP32K9.	Yes	No	No default	

Parameter: Profiles History database

Description

Create and Upgrade the Profiles History database to track profiles history in DB2 10 and later releases.

This parameter is optional.

Steps and parameters for specifying the Profiles History database

Step or parameter	Required?	Discovered?	Default value	Your value
Profiles History database Specify YES to Create and Upgrade the Profiles History database. Specify NO to not Create and Upgrade the Profiles History database.	Yes	Yes	Yes	

Parameter: Reverse engineering objects

Description

Defines the reverse engineering stored procedure, ADB2RE, and the required temporary tables so that you can use reverse engineering from additional software products such as Control Center OS/390.

To use Reverse Engineering from other software products such as Control Center OS/390, generate and submit the Reverse Engineering job template ADBREST. ADBREST creates the reverse engineering stored procedure ADB2RE. The ADB2RE stored procedure must be defined with the SECURITY USER clause and must run in a WLM-managed stored procedure address space.

This parameter is optional.

Steps and parameters for specifying Reverse Engineering objects

Step or parameter	Required?	Discovered?	Default value	Your value
Reverse Engineering objects Specify YES so that Reverse Engineering objects are defined. Specify NO so that Reverse Engineering objects are not defined.	Yes	Yes	Yes	

Parameter: Stored procedure ADB2RCP

Description

Customizes JCL that is used to create the stored procedure for running DB2 commands when you are connected to a remote site.

This parameter is optional.

Steps and parameters for specifying the ADB2RCP stored procedure

Step or parameter	Required?	Discovered?	Default value	Your value
Stored procedure ADB2RCP Specify YES to create and bind the ADB2RCP stored procedure. Specify NO to not create this stored procedure.	Yes	Yes	Yes	

Parameter: VIEW RUNSTATS objects

Description

Creates views that allow the creators to update the RUNSTATS information for their own objects in the catalog.

This parameter is optional.

Steps and parameters for specifying VIEW RUNSTATS objects

Step or parameter	Required?	Discovered?	Default value	Your value
VIEW RUNSTATS objects Specify YES to create views of RUNSTAT objects. Specify NO to not create views of RUNSTAT objects.	Yes	Yes	Yes	

Parameter: Indexes

Description

Creates additional indexes to improve performance for DB2 Admin Tool. This task depends on the value of the **Level Number** field that is specified on the DB2 Parameters panel.

This parameter is optional.

Steps and parameters for specifying indexes

Step or parameter	Required?	Discovered?	Default value	Your value
Indexes Specify YES to create additional indexes. Specify NO to not create additional indexes.	Yes	Yes	Yes	

Parameter: GRANT on DB2 catalog tables

Description

Gives GRANT Privilege on DB2 catalog tables. This task depends on the value of the **Level Number** field that is specified on the DB2 Parameters panel.

This parameter is optional.

Steps and parameters for specifying GRANT on DB2 catalog tables

Step or parameter	Required?	Discovered?	Default value	Your value
GRANT privilege on DB2 Catalog Tables Specify YES to give GRANT Privilege on DB2 catalog tables. Specify NO to not give GRANT Privilege on DB2 catalog tables.	Yes	Yes	Yes	

Task: Bind Plans and Packages

Description
> Binds plans and packages.

> This task is required.

Jobs generated
> ADBBIN*ab*, where *ab* are alphanumeric characters that are assigned by Tools Customizer. This job is based on the ADBBIND template and is in the *job_sequence_number*BIND*DB2_entry_ID* member.

Required authority
> The user ID that runs the job must have SYSADM or equivalent authority.

Steps and parameters for the Bind Plans and Packages task

Step or parameter	Required?	Discovered?	Default value	Your value
BIND OWNER Specify the Authorization ID of the Plan and Package owner.	Yes	No	No default	

Task: Sample JCL for ADBTEP2 execution

Description
> Tests ADBTEP2.

> This task is optional.

Jobs generated
> ADBTEP*ab*, where *ab* are alphanumeric characters that are assigned by Tools Customizer. This job is based on the ADBTEP2R template and is in the *job_sequence_number*TEP2*DB2_entry_ID* member.

Required authority
> The user ID that runs the job must have *authority*.

Steps and parameters for the Sample JCL for ADBTEP2 execution task

Step or parameter	Required?	Discovered?	Default value	Your value
Run ADBTEP2 After inserting SQL or other commands in the SYSIN DD, submit this job to test program ADBTEP2.	Yes	No	Selected	

Task: InfoSphere OCM integration enablement

Description
> Manages enablement of OCM.

> This task is optional.

Jobs generated

This task generates the following jobs:

- ADBCFG*ab*, where *ab* are alphanumeric characters that are assigned by Tools Customizer. This job is based on the ADBCFGBD template and is in the *job_sequence_number*CFGB*DB2_entry_ID* member.
- ADBCFG*ab*, where *ab* are alphanumeric characters that are assigned by Tools Customizer. This job is based on the ADBCFGPM template and is in the *job_sequence_number*CFGP*DB2_entry_ID* member.
- ADBLIM*ab*, where *ab* are alphanumeric characters that are assigned by Tools Customizer. This job is based on the ADBLIM template and is in the *job_sequence_number*LIM*DB2_entry_ID* member.

Required authority

The user ID that runs these jobs must have SYSADM or equivalent authority.

Steps and parameters for the InfoSphere OCM integration enablement task

Step or parameter	Required?	Discovered?	Default value	Your value
Bind DB2 Admin package on OCM repository database Bind DB2 Admin package on OCM repository database.	No	No	Not selected	
DB2 location name DB2 location name of the OCM repository database. Leave blank if local. Specify the value that is defined in the LOCATION column of the SYSIBM.LOCATIONS table in your DB2 catalog.	No	No	No default	
Set DB2 Admin settings for integration with OCM Deploy settings for OCM integration.	No	No	Not selected	
Enable recording to OCM Specify YES to enable DB2 Admin to store information about schema and authorization changes that are implemented by using DB2 Admin Tool Change Management.	No	No	NO	
Action to take on error Action to take when an error occurs while attempting to store data into the OCM repository: Specify STOP to issue an error message and stop processing. Specify LOCAL to attempt to store the data into local backup tables for OCM. If data cannot be stored locally, DB2 Admin stops processing. Specify OVERRIDE to do the same as option LOCAL, except the user is allowed to specify an override that continues DB2 Admin processing even if the data cannot be stored locally.	No	No	LOCAL	

Steps and parameters for the InfoSphere OCM integration enablement task

Step or parameter	Required?	Discovered?	Default value	Your value
Sample job to move data from local system to OCM repository database Sample job to run the ADBLIM program	No	No	Not selected	

Task: Installation verification jobs

Description
> Tests CM Batch.
>
> This task is optional.

Jobs generated
> ADBCMB*ab*, where *ab* are alphanumeric characters that are assigned by Tools Customizer. This job is based on the ADBCMBIV template and is in the *job_sequence_number*CMBI*DB2_entry_ID* member.

Required authority
> The user ID that runs the job must have SYSADM or equivalent authority.

Steps and parameters for the Installation verification jobs task

Step or parameter	Required?	Discovered?	Default value	Your value
Generate a CM Batch verification job Use this option to generate a batch job to verify that the CM Batch JCL procedure works.	No	No	Not selected	

LPAR Parameters section

Description
> This section contains LPAR parameters. All parameters are required. During the customization process, you will enter these values on the LPAR Parameters panel (CCQPLPR).

ISPF Libraries

Parameter	Required?	Discovered?	Default value	Your value
Message library The data set of the ISPF Message library that is used by batch jobs that are generated by Tools Customizer, where applicable. Enter a fully qualified valid data set name. You can specify multiple values for this parameter.	Yes	Yes	No default	

ISPF Libraries

Parameter	Required?	Discovered?	Default value	Your value
Panel library The data set of the ISPF Panel library that is used by batch jobs that are generated by Tools Customizer, where applicable. Enter a fully qualified valid data set name. You can specify multiple values for this parameter.	Yes	Yes	No default	
Skeleton library The data set of the ISPF Skeleton library that is used by batch jobs that are generated by Tools Customizer, where applicable. Enter a fully qualified valid data set name. You can specify multiple values for this parameter.	Yes	Yes	No default	
Table library The data set of the ISPF Table library that is used by batch jobs that are generated by Tools Customizer, where applicable. Enter a fully qualified valid data set name. You can specify multiple values for this parameter.	Yes	Yes	No default	
Load library Enter the ISPF Load library. This library is used by the Tools Customizer generated batch jobs, where applicable. Specifying the ISPF Load library is optional, if ISPF is already available in Link List.	Yes	No	No default	

Other Parameters

Parameter	Required?	Discovered?	Default value	Your value
Unit name for TSO work data sets The unit name for the TSO work data sets. The name must be 8 characters or less.	Yes	Yes	WDS	
Unit name for batch work data sets The unit name for the batch work data sets. The name must be 8 characters or less.	Yes	Yes	WDS	

Other Parameters

Parameter	Required?	Discovered?	Default value	Your value
Unicode translation technique The technique for Unicode translation.	Yes	Yes	UTF-8	

DB2 Parameters section

Description

    This section contains DB2 parameters. All parameters are required. During the customization process, you will enter these values on the DB2 Parameters panel (CCQPDB2).

DB2 parameters

Parameter	Required?	Discovered?	Default value	Your value
DB2 subsystem ID The name of the DB2 subsystem, which is also called the SSID. The value must be 4 characters or less.	Yes	Yes	No default	
Group attach name The name of the group attach name.	Yes	Yes	No default	
Started task name for MSTR services The name to start the DB2 subsystem system services. The value must be 8 characters or less.				

General DB2 Information

Parameter	Required?	Discovered?	Default value	Your value
Mode The mode in which the DB2 subsystem is running. The following values are valid: • CM is compatibility mode on all listed DB2 versions except DB2 10. • CM8 is conversion mode from DB2 V8 on DB2 10. • CM9 is conversion mode from DB2 Version 9.1 on DB2 10. • NFM is new-function mode on all listed DB2 versions.	Yes	Yes	NFM	

General DB2 Information

Parameter	Required?	Discovered?	Default value	Your value
Level number The version, release, and modification level of the DB2 subsystem. The following values are valid: • 910 is valid only for CM or NFM. • 101 is valid only for CM8, CM9 or NFM. • 111 is valid only for CM or NFM.	Yes	Yes	blank	

DB2 Utilities

Parameter	Required?	Discovered?	Default value	Your value
Plan name for the DSNTEP2 utility The plan name for the DSNTEP2 utility. The value must be 8 characters or less.	Yes	Yes	DSNTEP2	
Plan name for the DSNTIAD utility The plan name for the DSNTIAD utility. The value must be 8 characters or less.	Yes	Yes	DSNTIAD	

DB2 Admin Subsystem Parameters

Parameter	Required?	Discovered?	Default value	Your value
DB2 subsystem description A description for the DB2 subsystem. The value must be 72 characters or less.	Yes	Yes	No default	
Type of DB2 security exit The type of DB2 security exit that is installed for the DB2 subsystem. Valid values are: • STD: Standard DB2 security exit (default) • SAMPLE: Sample DB2 security exit (logic being simulated) • AUTH: Local DB2 security exit that must run authorized • NOCALL: Do not call the security exit. DB2 Admin Tool cannot show SQL IDs. • OWN: Local DB2 security exit that can run unauthorized. Or leave this input field blank to use the "Type of DB2 security exit" setting from the Product Parameters panel.	Yes	Yes	No default	

DB2 Admin Subsystem Parameters

Parameter	Required?	Discovered?	Default value	Your value
Enable DB2 Cloning Tool Launch DB2 Cloning Tool from within DB2 Administration Tool as an optional choice for migrating objects/data. Select YES to enable this option, NO to disable, or leave blank to use the Product Parameter default setting.	Yes	Yes	No default	
Cloning Tool CLIST lib Specify the CLIST library that contains the DB2 Cloning Tool invocation CLIST.	Yes	Yes	No default	
Enable DB2 Table Editor Launch DB2 Table Editor from within DB2 Administration Tool as an optional choice to quickly access, update, and delete data. Select YES to enable this option, NO to disable, or leave blank to use the Product Parameter default setting.	Yes	Yes	No default	
Table Editor CLIST(mbr) Specify the startup clist used to start the DB2 Table Editor. For example: *hlvlqual*.SETISAMP(ETI)	Yes	Yes	No default	
JOB class for DB2 utilities Default Job class to be used for running DB2 utilities. Enter a valid value of 1 character.	Yes	Yes	No default	
SYSAFF for DB2 utilities The SYSAFF job parameter to be used for batch DB2 Utility jobs. This parameter ensures that batch DB2 Utility jobs are run on the same operating system as the DB2 subsystem. Enter a valid value of 1 - 4 characters.	Yes	Yes	No default	
DB2 Admin APF library Used for: (1)Authorization Switching when building ALTER JCL, and (2)Modules ADB2ATH and ADB2UTIL that otherwise should be in the link list.	Yes	Yes	No default	

DB2 Admin Subsystem Parameters

Parameter	Required?	Discovered?	Default value	Your value
System identification method The system identification method is used to make sure batch utility jobs that are created with DB2 Admin execute on the same MVS system as the DB2 subsystem. This is done by placing a /*JOBPARM SYSAFF line in the JCL. Valid values are: SMFID (use SMF ID, only valid if SMF is active); JESID (use JES ID, only valid on JES2 systems); NONE (do not include a /*JOBPARM SYSAFF card in the generated JCL); SYSNAME (use MVS system name from CVT control block); or name (use name as SYSAFF name).	Yes	Yes	No default	
Installation name The installation name is a text string that is carried forward by DB2 Admin and can be used in local modifications.	Yes	Yes	No default	
Utility data set prefix High-level qualifier (HLQ) of the data sets that are used in DB2 utility jobs. Valid values are: USERID, OWNER, CREATEDBY, or name (use name as HLQ).	Yes	Yes	No default	
Remote DB2 subsystem name DB2 subsystem name of the remote DB2 subsystem. Leave blank if local.	Yes	Yes	No default	
Remote DB2 location name DB2 location name of the remote DB2 subsystem. Leave blank if local. Specify the value that is defined in the LOCATION column of the SYS1.LOCATIONS table in your DB2 catalog.	Yes	Yes	No default	
Enable authorization switching Specify YES to enable the Authorization Switching function for the current DB2 subsystem. Specify NO to disable Authorization Switching.	Yes	Yes	No default	
Authorization switching ID DB2 Security ID to use for auth-switching	Yes	Yes	No default	

DB2 Admin Subsystem Parameters

Parameter	Required?	Discovered?	Default value	Your value
ISPF application ID Identifies the member name in which the ISPF profile variables are saved for the DB2 Administration Tool. The default value is null with an application ID of ISR. If you use a minus sign with this parameter, the value set for this parameter is overridden by the DB2 Administration Tool, which is ISR.	Yes	Yes	No default	
PROMPT Options The installation default value for Prompt Options. Specify YES or No.	Yes	Yes	No default	
Reset to defaults at startup The installation default value for the Reset to Default at Startup parameter. Specify YES or No.	Yes	Yes	No default	
Number of DSNUPROC procedure job steps Subsystem default number of job steps in the DSNUPROC procedure.	Yes	Yes	No default	
Allow switch of SSID Allows switch of SSID for DB2 subsystems. Specify YES or No.	Yes	Yes	No default	
DB2 CONCENTRATE STATEMENTS WITH LITERALS Use the DB2 CONCENTRATE STATEMENTS WITH LITERALS attribute on all dynamic SQL statements. The default is YES. Valid only with DB2 V10 or higher.	Yes	Yes	No default	
DB2 use CONCURRENT clause on SQL Use the DB2 CONCURRENTLY COMMITTED attribute on all dynamic SQL statements. The default is YES. Valid only with DB2 V10 or higher.	Yes	Yes	No default	
User cmds lib(mbr) User commands library and member.	Yes	Yes	No default	
Automatic deletion of compare results Enter "YES" if you want to automatically delete saved compare results as part of the DB2 Administration Tool's cleansing process.	Yes	Yes	No default	

DB2 Admin Subsystem Parameters

Parameter	Required?	Discovered?	Default value	Your value
High Performance Unload (HPU) enabled Enter "YES" if you want to use HPU for Unloads for a specific subsystem. Specifying NO disables this support.	Yes	Yes	No default	
HPU load library The data set name for the High Performance Unload (HPU) SINZLINK load library when HPU is enabled. This variable is ignored if HPU is not enabled. Do not specify the HPU SINZLOAD data set, since this may cause an abend because of APF-authorization issues.	Yes	Yes	No default	
HPU parameter library The data set name for the High Performance Unload (HPU) SINZPARM parm library when HPU is enabled. This variable is ignored if HPU is not enabled. Do not specify the HPU SINZLOAD data set, since this may cause an abend because of APF-authorization issues.	Yes	Yes	No default	
REXX user exit lib The data set names for the REXX user exits used to specify overwrite values for masking fields DSSIZE, PRIQTY, SECQTY, DEFER, and DEFINE.	Yes	Yes	No default	

Create Checkpoint table parameters

Parameter	Required?	Discovered?	Default value	Your value
Owner name Used by SET CURRENT SQLID to set the owner name upon creation of the database objects. Enter a valid value of 1 to 128 characters.	Yes	Yes	No default	
Database name Name of the database where the objects and data will be stored. Enter a valid value of 1 - 8 characters.	Yes	Yes	No default	
STOGROUP name The name of the Storage Group (STOGROUP) that will be used when creating the database objects. Enter a valid value of 1 - 8 characters.	Yes	Yes	No default	

Create Checkpoint table parameters

Parameter	Required?	Discovered?	Default value	Your value
STOGROUP volumes Defines the volumes of the STOGROUP that will be used when creating the database objects. Enter a list of one or more VOLSERs separated by commas. Maximum input field length is 128 characters.	Yes	Yes	No default	
STOGROUP VCAT A catalog name used to identify the VSAM Catalog (VCAT) for the STOGROUP. Enter a valid value of 1 - 8 characters.	Yes	Yes	No default	
Tablespace name prefix The table space objects that will be created with a name prefixed with 1 - 6 characters.	Yes	Yes	No default	
Tablespace BUFFERPOOL name The buffer pool to be used when creating the table space objects. Valid values are: BP0 - BP49, BP8K0 - BP8K9, BP16K0 - BP16K9, BP32K, BP32K1 - BP32K9.	Yes	Yes	No default	
Index BUFFERPOOL name The buffer pool to be used when creating the index objects. Valid values are: BP0 - BP49, BP8K0 - BP8K9, BP16K0 - BP16K9, BP32K, BP32K1 - BP32K9.	Yes	Yes	No default	

Change Management database parameters

Parameter	Required?	Discovered?	Default value	Your value
Owner name Used by SET CURRENT SQLID to set the owner name upon creation of the database objects. Enter a valid value of 1 to 128 characters.	Yes	Yes	No default	
Database name Name of the database where the objects and data will be stored. Enter a valid value of 1 - 8 characters.	Yes	Yes	No default	
STOGROUP name The name of the Storage Group (STOGROUP) that will be used when creating the database objects. Enter a valid value of 1 - 8 characters.	Yes	Yes	No default	

Change Management database parameters

Parameter	Required?	Discovered?	Default value	Your value
STOGROUP volumes Defines the volumes of the STOGROUP that will be used when creating the database objects. Enter a list of one or more OLSERs separated by commas. Maximum input field length is 128 characters.	Yes	Yes	No default	
STOGROUP VCAT A catalog name that is used to identify the VSAM Catalog (VCAT) for the STOGROUP. Enter a valid value of 1 - 8 characters.	Yes	Yes	No default	
Tablespace name prefix The table space objects that will be created with a name prefixed with 1 - 4 characters.	Yes	Yes	No default	
Tablespace BUFFERPOOL name The buffer pool to be used when creating the table space objects. Valid values are: BP0 - BP49, BP8K0 - BP8K9, BP16K0 - BP16K9, BP32K, BP32K1 - BP32K9.	Yes	No	No default	
Index BUFFERPOOL name The buffer pool to be used when creating the index objects. Valid values are: BP0 - BP49, BP8K0 - BP8K9, BP16K0 - BP16K9, BP32K, BP32K1 - BP32K9.	Yes	Yes	No default	
Enable Change Management Set this parameter to YES if you intend to use Change Management for this particular subsystem.	Yes	Yes	YES	
Enable Allow Change Delete Enable the delete change line command, but only if the Change Management database objects owner is not blank.	Yes	Yes	No default	
One PROCLIB for multiple SSIDs Specify whether the CM Batch procedure is to support multiple DB2 subsystems. If NO, the procedure name will be GOC(SSID). Otherwise, the procedure name that is specified on the Product Parameter panel is used.	Yes	Yes	No default	

Create Catalog Copy Version Table parameters

Parameter	Required?	Discovered?	Default value	Your value
Owner name Used by SET CURRENT SQLID to set the owner name upon creation of the database objects. Enter a valid value of 1 to 128 characters.	Yes	Yes	No default	
Database name Name of the database where the objects and data will be stored. Enter a valid value of 1 - 8 characters.	Yes	Yes	No default	
STOGROUP name The name of the Storage Group (STOGROUP) that will be used when creating the database objects. Enter a valid value of 1 - 8 characters.	Yes	Yes	No default	
STOGROUP volumes Defines the volumes of the STOGROUP that will be used when creating the database objects. Enter a list of one or more VOLSERs separated by commas. Maximum input field length is 128 characters.	Yes	Yes	No default	
STOGROUP VCAT A catalog name that is used to identify the VSAM Catalog (VCAT) for the STOGROUP. Enter a valid value of 1 - 8 characters.	Yes	Yes	No default	
Tablespace name prefix The table space objects that will be created with a name prefixed with 1 - 6 characters.	Yes	Yes	No default	

Create Profiles History database parameters

Parameter	Required?	Discovered?	Default value	Your value
Owner name Used by SET CURRENT SQLID to set the owner name upon creation of the Profiles History objects. Enter a valid value of 1 to 128 characters.	Yes	Yes	No default	
Current schema Used by SET CURRENT SCHEMA to establish an implicit qualifier to be used when creating the database objects. Enter a valid value of 1 to 128 characters.	Yes	Yes	No default	

Create Profiles History database parameters

Parameter	Required?	Discovered?	Default value	Your value
Database name Database name where the Profiles History objects will be stored. Enter a valid value of 1 - 8 characters.	Yes	Yes	No default	
STOGROUP name The name of the Storage Group (STOGROUP) that will be used when creating the database objects. Enter a valid value of 1 - 8 characters.	Yes	Yes	No default	
STOGROUP volumes Defines the volumes of the STOGROUP which will be used when creating the database objects. Enter a list of one or more VOLSERs separated by commas. Maximum input field length is 128 characters.	Yes	Yes	No default	
STOGROUP VCAT A catalog name used to identify the VSAM Catalog (VCAT) for the STOGROUP. Enter a valid value of 1 - 8 characters.	Yes	Yes	No default	
Tablespace name prefix The table space objects that will be created with a name prefixed with 1 - 6 characters.	Yes	Yes	No default	

DB2 Libraries parameters

Parameter	Required?	Discovered?	Default value	Your value
DB2 load library Enter the DB2 load library SDSNLOAD and DB2 exit library SDSNEXIT. You can specify multiple values for this parameter.	Yes	Yes	No default	
DB2 run library The data set name of the DB2 run library. You can specify multiple values for this parameter.	Yes	Yes	No default	
DB2 message library The data set name of the DB2 message library. You can specify multiple values for this parameter.	Yes	Yes	No default	

DB2 Libraries parameters

Parameter	Required?	Discovered?	Default value	Your value
DB2 panel library The data set name of the DB2 panel library. You can specify multiple values for this parameter.	Yes	Yes	No default	
DB2 skeleton library The data set name of the DB2 skeleton library. You can specify multiple values for this parameter.	Yes	Yes	No default	
DB2 table library The data set name of the DB2 table library. You can specify multiple values for this parameter.	Yes	Yes	No default	
DB2 CLIST library The data set name of the DB2 CLIST library. You can specify multiple values for this parameter.	Yes	Yes	No default	

DB2 Admin Tool Libraries parameters

Parameter	Required?	Discovered?	Default value	Your value
DB2 Admin Tool load library The data set name of the DB2 Admin Tool load library. You can specify multiple values for this parameter.	Yes	Yes	No default	
Admin Tool SADBMLIB The data set name of the DB2 Admin Tool message library. You can specify multiple values for this parameter.	Yes	Yes	No default	
Admin Tool SADBPLIB The data set name of the DB2 Admin Tool panel library. You can specify multiple values for this parameter.	Yes	Yes	No default	
Admin Tool SADBSLIB The data set name of the DB2 Admin Tool skeleton library. You can specify multiple values for this parameter.	Yes	Yes	No default	

DB2 Admin Tool Libraries parameters

Parameter	Required?	Discovered?	Default value	Your value
Admin Tool SADBTLIB The data set name of the DB2 Admin Tool table library. You can specify multiple values for this parameter.	Yes	Yes	No default	
DAdmin Tool SADBCLST The data set name of the DB2 Admin Tool CLIST library. You can specify multiple values for this parameter.	Yes	Yes	No default	
Admin Tool SADBEXEC The data set name of the DB2 Admin Tool REXX exec library. You can specify multiple values for this parameter.	Yes	Yes	No default	
Admin Tool SADBDBRM The data set name of the DB2 Admin Tool DBRM library. You can specify multiple values for this parameter.	Yes	Yes	No default	

DB2 Admin main menu - First Option parameters

Parameter	Required?	Discovered?	Default value	Your value
Option 1 Produces an additional option to display on the DB2 Administration Menu panel ADB2. Enter a valid value of 1 - 72 characters.	Yes	Yes	No default	
Option 1 description A description of the menu option to be displayed on the DB2 Administration Menu panel ADB2. Enter a valid value of 1 - 72 characters.	Yes	Yes	No default	
ISPF statement for option 1 The ISPF statement that the DB2 Administration Tool will execute for this menu option. Enter a valid value of 1 - 72 characters.	Yes	Yes	No default	
ISPF panel for option 1 The name of the ISPF panel that the DB2 Administration Tool will display for this menu option. Enter a valid value of 1 - 8 characters.	Yes	Yes	No default	

DB2 Admin main menu - First Option parameters

Parameter	Required?	Discovered?	Default value	Your value
SQL statement for option 1 The SQL statement that the DB2 Administration Tool will execute for this menu option. Enter a valid value of 1 - 256 characters.	Yes	Yes	No default	
DB2 Admin Tool command for option 1 The DB2 Administration Tool command for this menu option. Enter a valid value of 1 - 256 characters.	Yes	Yes	No default	
New DB2 attachment for option 1 Specify YES to start a new DB2 attachment for this menu option. Otherwise, specify NO.	Yes	Yes	No default	

DB2 Admin main menu - Second Option parameters

Parameter	Required?	Discovered?	Default value	Your value
Option 2 Produces an additional option to display on the DB2 Administration Menu panel ADB2. Enter a valid value of 1 - 72 characters.	Yes	Yes	No default	
Option 2 description A description of the menu option to be displayed on the DB2 Administration Menu panel ADB2. Enter a valid value of 1 - 72 characters.	Yes	Yes	No default	
ISPF statement for option 2 The ISPF statement that the DB2 Administration Tool will execute for this menu option. Enter a valid value of 1 - 72 characters.	Yes	Yes	No default	
ISPF panel for option 2 The name of the ISPF panel that the DB2 Administration Tool will display for this menu option. Enter a valid value of 1 - 8 characters.	Yes	Yes	No default	
SQL statement for option 2 The SQL statement that the DB2 Administration Tool will execute for this menu option. Enter a valid value of 1 - 256 characters.	Yes	Yes	No default	
DB2 Admin Tool command for option 2 The DB2 Administration Tool command for this menu option. Enter a valid value of 1 - 256 characters.	Yes	Yes	No default	

DB2 Admin main menu - Second Option parameters

Parameter	Required?	Discovered?	Default value	Your value
New DB2 attachment for option 2 Specify YES to start a new DB2 attachment for this menu option. Otherwise, specify NO.	Yes	Yes	No default	

DB2 Admin main menu - Third Option parameters

Parameter	Required?	Discovered?	Default value	Your value
Option 3 Produces an additional option to display on the DB2 Administration Menu panel ADB2. Enter a valid value of 1 - 72 characters.	Yes	Yes	No default	
Option 3 description A description of the menu option to be displayed on the DB2 Administration Menu panel ADB2. Enter a valid value of 1 - 72 characters.	Yes	Yes	No default	
ISPF statement for option 3 The ISPF statement that the DB2 Administration Tool will execute for this menu option. Enter a valid value of 1 - 72 characters.	Yes	Yes	No default	
ISPF panel for option 3 The name of the ISPF panel that the DB2 Administration Tool will display for this menu option. Enter a valid value of 1 - 8 characters.	Yes	Yes	No default	
SQL statement for option 3 The SQL statement that the DB2 Administration Tool will execute for this menu option. Enter a valid value of 1 - 256 characters.	Yes	Yes	No default	
DB2 Admin Tool command for option 3 The DB2 Administration Tool command for this menu option. Enter a valid value of 1 - 256 characters.	Yes	Yes	No default	
New DB2 attachment for option 3 Specify YES to start a new DB2 attachment for this menu option. Otherwise, specify NO.	Yes	Yes	No default	

DB2 Admin main menu - Fourth Option parameters

Parameter	Required?	Discovered?	Default value	Your value
Option 4 Produces an additional option to display on the DB2 Administration Menu panel ADB2. Enter a valid value of 1 - 72 characters.	Yes	Yes	No default	
Option 4 description A description of the menu option to be displayed on the DB2 Administration Menu panel ADB2. Enter a valid value of 1 - 72 characters.	Yes	Yes	No default	
ISPF statement for option 4 The ISPF statement that the DB2 Administration Tool will execute for this menu option. Enter a valid value of 1 - 72 characters.	Yes	Yes	No default	
ISPF panel for option 4 The name of the ISPF panel that the DB2 Administration Tool will display for this menu option. Enter a valid value of 1 - 8 characters.	Yes	Yes	No default	
SQL statement for option 4 The SQL statement that the DB2 Administration Tool will execute for this menu option. Enter a valid value of 1 - 256 characters.	Yes	Yes	No default	
DB2 Admin Tool command for option 4 The DB2 Administration Tool command for this menu option. Enter a valid value of 1 - 256 characters.	Yes	Yes	No default	
New DB2 attachment for option 4 Specify YES to start a new DB2 attachment for this menu option. Otherwise, specify NO.	Yes	Yes	No default	

DB2 Admin main menu - Fifth Option parameters

Parameter	Required?	Discovered?	Default value	Your value
Option 5 Produces an additional option to display on the DB2 Administration Menu panel ADB2. Enter a valid value of 1 - 72 characters.	Yes	Yes	No default	

DB2 Admin main menu - Fifth Option parameters

Parameter	Required?	Discovered?	Default value	Your value
Option 5 description A description of the menu option to be displayed on the DB2 Administration Menu panel ADB2. Enter a valid value of 1 - 72 characters.	Yes	Yes	No default	
ISPF statement for option 5 The ISPF statement that the DB2 Administration Tool will execute for this menu option. Enter a valid value of 1 - 72 characters.	Yes	Yes	No default	
ISPF panel for option 5 The name of the ISPF panel that the DB2 Administration Tool will display for this menu option. Enter a valid value of 1 - 8 characters.	Yes	Yes	No default	
SQL statement for option 5 The SQL statement that the DB2 Administration Tool will execute for this menu option. Enter a valid value of 1 - 256 characters.	Yes	Yes	No default	
DB2 Admin Tool command for option 5 The DB2 Administration Tool command for this menu option. Enter a valid value of 1 - 256 characters.	Yes	Yes	No default	
New DB2 attachment for option 5 Specify YES to start a new DB2 attachment for this menu option. Otherwise, specify NO.	Yes	Yes	No default	

DB2 Admin main menu - Sixth Option parameters

Parameter	Required?	Discovered?	Default value	Your value
Option 6 Produces an additional option to display on the DB2 Administration Menu panel ADB2. Enter a valid value of 1 - 72 characters.	Yes	Yes	No default	
Option 6 description A description of the menu option to be displayed on the DB2 Administration Menu panel ADB2. Enter a valid value of 1 - 72 characters.	Yes	Yes	No default	

DB2 Admin main menu - Sixth Option parameters

Parameter	Required?	Discovered?	Default value	Your value
ISPF statement for option 6 The ISPF statement that the DB2 Administration Tool will execute for this menu option. Enter a valid value of 1 - 72 characters.	Yes	Yes	No default	
ISPF panel for option 6 The name of the ISPF panel that the DB2 Administration Tool will display for this menu option. Enter a valid value of 1 - 8 characters.	Yes	Yes	No default	
SQL statement for option 6 The SQL statement that the DB2 Administration Tool will execute for this menu option. Enter a valid value of 1 - 256 characters.	Yes	Yes	No default	
DB2 Admin Tool command for option 6 The DB2 Administration Tool command for this menu option. Enter a valid value of 1 - 256 characters.	Yes	Yes	No default	
New DB2 attachment for option 6 Specify YES to start a new DB2 attachment for this menu option. Otherwise, specify NO.	Yes	Yes	No default	

DB2 Admin main menu - Seventh Option parameters

Parameter	Required?	Discovered?	Default value	Your value
Option 7 Produces an additional option to display on the DB2 Administration Menu panel ADB2. Enter a valid value of 1 - 72 characters.	Yes	Yes	No default	
Option 7 description A description of the menu option to be displayed on the DB2 Administration Menu panel ADB2. Enter a valid value of 1 - 72 characters.	Yes	Yes	No default	
ISPF statement for option 7 The ISPF statement that the DB2 Administration Tool will execute for this menu option. Enter a valid value of 1 - 72 characters.	Yes	Yes	No default	
ISPF panel for option 7 The name of the ISPF panel that the DB2 Administration Tool will display for this menu option. Enter a valid value of 1 - 8 characters.	Yes	Yes	No default	

DB2 Admin main menu - Seventh Option parameters

Parameter	Required?	Discovered?	Default value	Your value
SQL statement for option 7 The SQL statement that the DB2 Administration Tool will execute for this menu option. Enter a valid value of 1 - 256 characters.	Yes	Yes	No default	
DB2 Admin Tool command for option 7 The DB2 Administration Tool command for this menu option. Enter a valid value of 1 - 256 characters.	Yes	Yes	No default	
New DB2 attachment for option 7 Specify YES to start a new DB2 attachment for this menu option. Otherwise, specify NO.	Yes	Yes	No default	

DB2 Admin main menu - Eighth Option parameters

Parameter	Required?	Discovered?	Default value	Your value
Option 8 Produces an additional option to display on the DB2 Administration Menu panel ADB2. Enter a valid value of 1 - 72 characters.	Yes	Yes	No default	
Option 8 description A description of the menu option to be displayed on the DB2 Administration Menu panel ADB2. Enter a valid value of 1 - 72 characters.	Yes	Yes	No default	
ISPF statement for option 8 The ISPF statement that the DB2 Administration Tool will execute for this menu option. Enter a valid value of 1 - 72 characters.	Yes	Yes	No default	
ISPF panel for option 8 The name of the ISPF panel that the DB2 Administration Tool will display for this menu option. Enter a valid value of 1 - 8 characters.	Yes	Yes	No default	
SQL statement for option 8 The SQL statement that the DB2 Administration Tool will execute for this menu option. Enter a valid value of 1 - 256 characters.	Yes	Yes	No default	
DB2 Admin Tool command for option 8 The DB2 Administration Tool command for this menu option. Enter a valid value of 1 - 256 characters.	Yes	Yes	No default	

DB2 Admin main menu - Eighth Option parameters

Parameter	Required?	Discovered?	Default value	Your value
New DB2 attachment for option 8 Specify YES to start a new DB2 attachment for this menu option. Otherwise, specify NO.	Yes	Yes	No default	

DB2 Admin main menu - Ninth Option parameters

Parameter	Required?	Discovered?	Default value	Your value
Option 9 Produces an additional option to display on the DB2 Administration Menu panel ADB2. Enter a valid value of 1 - 72 characters.	Yes	Yes	No default	
Option 9 description A description of the menu option to be displayed on the DB2 Administration Menu panel ADB2. Enter a valid value of 1 - 72 characters.	Yes	Yes	No default	
ISPF statement for option 9 The ISPF statement that the DB2 Administration Tool will execute for this menu option. Enter a valid value of 1 - 72 characters.	Yes	Yes	No default	
ISPF panel for option 9 The name of the ISPF panel that the DB2 Administration Tool will display for this menu option. Enter a valid value of 1 - 8 characters.	Yes	Yes	No default	
SQL statement for option 9 The SQL statement that the DB2 Administration Tool will execute for this menu option. Enter a valid value of 1 - 256 characters.	Yes	Yes	No default	
DB2 Admin Tool command for option 9 The DB2 Administration Tool command for this menu option. Enter a valid value of 1 - 256 characters.	Yes	Yes	No default	
New DB2 attachment for option 9 Specify YES to start a new DB2 attachment for this menu option. Otherwise, specify NO.	Yes	Yes	No default	

DB2 Admin main menu - Tenth Option parameters

Parameter	Required?	Discovered?	Default value	Your value
Option 10 Produces an additional option to display on the DB2 Administration Menu panel ADB2. Enter a valid value of 1 - 72 characters.	Yes	Yes	No default	
Option 10 description A description of the menu option to be displayed on the DB2 Administration Menu panel ADB2. Enter a valid value of 1 - 72 characters.	Yes	Yes	No default	
ISPF statement for option 10 The ISPF statement that the DB2 Administration Tool will execute for this menu option. Enter a valid value of 1 - 72 characters.	Yes	Yes	No default	
ISPF panel for option 10 The name of the ISPF panel that the DB2 Administration Tool will display for this menu option. Enter a valid value of 1 - 8 characters.	Yes	Yes	No default	
SQL statement for option 10 The SQL statement that the DB2 Administration Tool will execute for this menu option. Enter a valid value of 1 - 256 characters.	Yes	Yes	No default	
DB2 Admin Tool command for option 10 The DB2 Administration Tool command for this menu option. Enter a valid value of 1 - 256 characters.	Yes	Yes	No default	
New DB2 attachment for option 10 Specify YES to start a new DB2 attachment for this menu option. Otherwise, specify NO.	Yes	Yes	No default	

Chapter 3. Customizing DB2 Admin

After DB2 Admin has been installed, complete the following steps to customize DB2 Admin.

The following topics assume that you have completed the installation instructions found in the *Program Directory for IBM DB2 Administration Tool for z/OS (GI10-8844)*.

For every DB2 subsystem on which you want to use DB2 Admin, you must run Tools Customizer to customize DB2 Admin. Before beginning this customization process, gather the names of all DB2 subsystems on which you want to run DB2 Admin.

It is recommended that you never modify the SMP/E target libraries or run jobs from the target libraries. Normally you create run time libraries based on these target libraries. There is also a mechanism in ADBL CLIST to allow you to integrate their modifications into a set of separate user libraries that are concatenated to the run time libraries. This way you never lose your modifications.

Examine these members in your user libraries against the new run time libraries to determine if you need to redo your modifications. You can do this manually by looking for differences or by installing a USERMOD so that SMP/E tracks your changes and notifies you. Another advantage of using USERMOD is that you can examine the new member and integrate the new lines of code into your customized version.

Topics:
- Roadmap: Customizing DB2 Admin for the first time
- "Roadmap: Migrating to DB2 Admin V11.2 from a previous release." on page 72
- "Roadmap: Recustomizing DB2 Admin 11.2" on page 75
- "Optional DB2 Admin customization tasks" on page 98

Roadmap: Customizing DB2 Admin for the first time

This roadmap lists and describes the steps for customizing DB2 Admin for the first time by using Tools Customizer.

Tip: Before you use this roadmap, complete the following worksheets to determine all of the customization values that you will need to supply during the customization process:
- "Worksheets: Gathering required data set names" on page 19
- Worksheets: Gathering parameter values for Tools Customizer

Complete the steps in the following table to customize DB2 Admin for the first time. A summary of each step is provided in the **Procedure** column, and links to detailed instructions and specific sections of the worksheets are provided in the **Links to more information** column, where applicable.

Table 1. Steps for customizing DB2 Admin for the first time

Step	Procedure	Links to more information

Start Tools Customizer.	1. Edit the CCQTCZ member in the *hlq*.TCZ110.SCCQEXEC data set.
	2. Locate `TCZHLQ="<TCz HLQ>"`.
	3. Change `"<TCz HLQ>"` to the high-level qualifier of your Tools Customizer EXEC data set, as shown in the following example:
	`TCZHLQ="`*hlq*`.TCZ110"`
	4. Save your changes.
	5. On the ISPF Command shell panel, issue the following command:
	`EX '`*hlq*`.TCZ110.SCCQEXEC(CCQTCZ)'`
		Detailed instructions:
		
		"Starting Tools Customizer" on page 76

Modify Tools Customizer settings.	1. On the CCQPHME panel, specify option 0 **User settings for Tools Customizer**. 2. Refer to the worksheets that you completed to specify values for the following required sections: • Customization library qualifier • Use DB2 group attach name • Metadata library • Discover output data set • Data store data set • User job card settings 3. Save your changes, and press Enter.	**Detailed instructions:** Settings for Tools Customizer **Worksheet:** "Modifying Tools Customizer user settings" on page 77
Specify the DB2 Admin Metadata Library.	1. On the CCQPHME panel, specify option 1 **Customize a product**. 2. On the CCQPHLQ panel, enter the following in the **Product or pack metadata library** field: DMTOOL.ADB2PAR.SADBDENU.	**Detailed instructions:** "Specifying the metadata library for the product to customize" on page 81
Create DB2 entries.	1. On the CCQPWRK panel, issue the ASSOCIATE primary command, and press Enter. 2. On the CCQPDAD panel, issue the CREATE primary command, and press Enter. 3. On the CCQPDCR panel, specify the information for the new DB2 entry, and press Enter. 4. On the CCQPDAD panel, issue the A line command against the new DB2 entry, and press Enter.	**Detailed instructions:** "Creating and associating DB2 entries" on page 84

Table 1. Steps for customizing DB2 Admin for the first time (continued)

Step	Procedure	Links to more information
Define product parameters.	1. On the CCQPWRK panel, specify the E line command against the **Product parameters** field. 2. Specify values for the following required sections on the CCQPPRD panel. For more information, refer to the worksheets that you completed. • Required parameters • Task: General customization • Task: Admin Tool Setup - Create and Upgrade • Task: Bind Plans and Packages • Task: Installation verification jobs **Important:** These are the minimum values to be specified. Select additional tasks and steps and define additional parameters to match your environment. 3. Press Enter to save and exit.	**Detailed instructions:** "Defining DB2 Admin parameters" on page 86 **Worksheet:** • Worksheets: Gathering parameter values for Tools Customizer • Task: General customization • Task: Admin Tool Setup - create or upgrade • Task: Bind Plans and Packages • Task: Installation verification jobs
Define LPAR parameters.	1. On the CCQPWRK panel, specify the E line command against the **LPAR parameters** field. 2. Specify values for the following required sections on the CCQPLPR panel. For more information, refer to the worksheets that you completed. • ISPF Libraries • Other Parameters • Change Management database **Important:** These are the minimum values to be specified. Select additional tasks and steps and define additional parameters to match your environment. 3. Press Enter to save and exit.	**Detailed instructions:** "Defining LPAR parameters" on page 87 **Worksheet:** LPAR Parameters section

Table 1. Steps for customizing DB2 Admin for the first time (continued)

Step	Procedure	Links to more information

Edit the DB2 entry.	1. On the CCQPWRK panel, issue the E line command against the new DB2 entry to edit the following parameters.
	
	2. Specify values for the following required sections on the CCQPDB2 panel. For more information, refer to the worksheets that you completed. For field-specific information, put the cursor in the input field and press F1.
	  • Mode
	  • Level
	  • DB2 subsystem description
	  • Remote DB2 subsystem name
	  • All parameters in the Create Checkpoint table section
	  • All parameters in the Change Management database section
	  • All parameters in the DB2 Libraries section
	  • All parameters in the DB2 Admin main menu - First Option and DB2 Admin main menu - Second Option section
	
	Important:
	  • These are the minimum values to be specified. Define additional parameters to match your environment.
	  • Some of the parameters on the CCQPDB2 panel are identical to parameters on the CCQPPRD panel. If you leave these parameters blank on the CCQPDB2 panel, Tools Customizer will use the values specified on the CCQPPRD panel. If you use unique values for specific DB2 entries, specify these values on the CCQPDB2 panel. For example, if five DB2 V11 subsystems use the STD DB2 security exit, specify STD on the CCQPPRD panel and leave the field blank on the CCQPDB2 panel for each subsystem.
	
	3. Press Enter to save and exit.
		Detailed instructions:
		
		  "Defining DB2 parameters" on page 89
		
		Worksheet:
		
		  DB2 Parameters section

Generate the jobs.	On the CCQPWRK panel, issue the G line command against the new DB2 entry, and press Enter.	**Detailed instructions:** "Generating customization jobs" on page 91
Submit the jobs.	On the CCQPCST panel, issue the E line command against the *ab*CUST*xy* member. **Important:** These are the minimum jobs to be submitted.	**Detailed instructions:** "Submitting customization jobs" on page 91

Roadmap: Migrating to DB2 Admin V11.2 from a previous release.

This roadmap lists and describes the steps for migrating to DB2 Admin V11.2 from a previous version, such as V11.1 or V10.2.

Tip: Before you use this roadmap, complete the following worksheets to determine all of the customization values that you will need to supply during the customization process:

- "Worksheets: Gathering required data set names" on page 19
- Worksheets: Gathering parameter values for Tools Customizer

Complete the steps in the following table to migrate to DB2 Admin V11.2 from a previous release. A summary of each step is provided in the **Procedure** column, and links to detailed instructions and specific sections of the worksheets are provided in the **Links to more information** column, where applicable.

Table 2. Steps for migrating to DB2 Admin V11.2 from a previous release

Step	Procedure	Links to more information
Start Tools Customizer.	1. On the ISPF Command shell panel, issue the following command: `EX 'hlq.TCZ110.SCCQEXEC(CCQTCZ)'`	**Detailed instructions:** "Starting Tools Customizer" on page 76
Specify the DB2 Admin Metadata Library.	1. On the CCQPHME panel, specify option 1 **Customize a product**. 2. On the CCQPHLQ panel, specify the DB2 Admin metadata library in the **Product or pack metadata library** field, such as DMTOOL.ADBB2PAR.SADBDENU.	**Detailed instructions:** "Specifying the metadata library for the product to customize" on page 81
Run the Discover EXEC.	1. On the CCQPWRK panel, specify DISCOVER on the command line. 2. Specify values for the following required input fields: • Source Customized table library (DMTOOL.ADBB1PAR.SADBTLIB) • Target Customized table library (DMTOOL.ADBB2PAR.SADBTLIB) **Tip:** For guidance on these input fields, position your cursor on the input field and press F1 (Help). 3. Issue the RUN primary command. Upon completion, the DISCOVER process will populate all Tools Customizer input fields from the previous customization to the current customization.	**Detailed instructions:** "Discovering DB2 Admin information automatically" on page 82 **Worksheet:** Customization values for the Discover EXEC

Step	Procedure	Links to more information

Define product parameters.	1. On the CCQPWRK panel, specify the E line command against the **Product parameters** field.
	
	2. Confirm that the DISCOVER process populated the input fields accordingly. The following input fields will need to be verified to ensure that the current DB2 Admin Tool libraries and Object Comparison libraries are specified:
	  • Admin Tool/OC CLIST
	  • Admin Tool DBRM
	  • Admin Tool/OC EXEC
	  • Admin Tool/OC Load
	  • Admin Tool/OC Message
	  • Admin Tool/OC Panel
	  • Admin Tool/OC Skeleton
	  • Admin Tool/OC Table
	  • Customized Table lib
	  • Admin Tool *hlq*
	  • CM Batch PROCLIB
		Detailed instructions:
		  "Defining DB2 Admin parameters" on page 86
		
		Worksheet:
		  • Worksheets: Gathering parameter values for Tools Customizer
		  • Task: General customization
		  • Task: Admin Tool Setup - create or upgrade
		  • Task: Installation verification jobs

Define LPAR parameters.	1. On the CCQPWRK panel, specify the E line command against the **LPAR parameters** field.
	
	2. Confirm that the DISCOVER process populated the input fields accordingly.
		Detailed instructions:
		  "Defining LPAR parameters" on page 87
		
		Worksheet:
		  LPAR Parameters section

Edit the new DB2 entry.	1. On the Customizer Workplace panel (CCQPWRK), enter the E line command against the site-specific SSID to display the DB2 Parameters panel (CCQPDB2).
	
	2. Confirm that the DISCOVER process populated the input fields accordingly.
		Detailed instructions:
		  "Defining DB2 parameters" on page 89
		
		Worksheet:
		  DB2 Parameters section

Generate the jobs.	On the CCQPWRK panel, issue the G line command against a site-specific SSID, and press Enter.	**Detailed instructions:** "Generating customization jobs" on page 91

Table 2. Steps for migrating to DB2 Admin V11.2 from a previous release (continued)

Step	Procedure	Links to more information
Submit the jobs.	On the CCQPCST panel, submit the generated jobs in the order they are displayed. For example, the following three generated jobs are all that need to be submitted to install or migrate DB2 Admin Tool for subsystem DSNB: ``` Cmd Member SSID GrpAttch Template Date Description - -------- ---- ---- -------- ---------- --------------------------> A0CUSTAA DSNB -- ADBCUST 2015/10/09 General customization A2SETUAA DSNB -- ADBSETUP 2015/10/09 Admin Tool Setup Task A5BINDAA DSNB -- ADBBIND 2015/10/09 Binds -----------------------End of customized jobs----------------------------- ``` If the Product Parameters option "Change Management database" is set to YES, the following jobs are generated and must be submitted in the order they are displayed: ``` Cmd Member SSID GrpAttch Template Date Description - -------- ---- ---- -------- ---------- --------------------------> A0CUSTAA DSNB -- ADBCUST 2015/10/09 General customization A2SETUAA DSNB -- ADBSETUP 2015/10/09 Admin Tool Setup Task A3CMBAT -- -- ADBCMBAT 2015/10/09 Create CM Batch JCL procedure A4CMBSAA DSNB -- ADBCMBSS 2015/10/09 Create CM Batch items A5BINDAA DSNB -- ADBBIND 2015/10/09 Binds B0CMBIAA DSNB -- ADBCMBIV 2015/10/09 Verify CM Batch JCL procedure -------------------------End of customized jobs---------------------------- ```	**Detailed instructions:** "Submitting customization jobs" on page 91

Roadmap: Recustomizing DB2 Admin 11.2

This roadmap lists and describes the steps for recustomizing DB2 Admin V11.2 by changing parameter values and generating new customization jobs.

Tip: Before you use this roadmap, complete the following worksheets to determine all of the customization values that you will need to supply during the customization process:

- "Worksheets: Gathering required data set names" on page 19
- Worksheets: Gathering parameter values for Tools Customizer

Complete the steps in the following table to recustomize DB2 Admin. A summary of each step is provided in the **Procedure** column, and links to detailed instructions and specific sections of the worksheets are provided in the **Links to more information** column, where applicable.

Table 3. Steps for recustomizing DB2 Admin V11.2

Step	Procedure	Links to more information
Start Tools Customizer.	1. On the ISPF Command shell panel, issue the following command: EX '*hlq*.TCZ110.SCCQEXEC(CCQTCZ)'	**Detailed instructions:** "Starting Tools Customizer" on page 76

Table 3. Steps for recustomizing DB2 Admin V11.2 (continued)

Step	Procedure	Links to more information
Specify the DB2 Admin Metadata Library.	1. On the CCQPHME panel, specify option 1 **Customize a product**. 2. On the CCQPHLQ panel, specify the DB2 Admin metadata library in the **Product or pack metadata library** field, such as DMTOOL.ADBB2PAR.SADBDENU.	**Detailed instructions:** "Specifying the metadata library for the product to customize" on page 81
Define product parameters, LPAR parameters, or DB2 parameters.	1. On the CCQPWRK panel, specify the E line command against the **Product parameters** field, the **LPAR parameters** field, or a DB2 entry, and press Enter. 2. Edit the specific tasks, steps, or parameters that you want to change. 3. Press Enter to save and exit.	**Detailed instructions:** • "Defining DB2 Admin parameters" on page 86 • "Defining LPAR parameters" on page 87 • "Defining DB2 parameters" on page 89
Generate the jobs.	On the CCQPWRK panel, issue the G line command against a site-specific SSID, and press Enter.	**Detailed instructions:** "Generating customization jobs" on page 91
Submit the jobs.	On the CCQPCST panel, submit the generated jobs in the order they are displayed.	**Detailed instructions:** "Submitting customization jobs" on page 91

Starting and preparing Tools Customizer for use

Use the provided REXX EXEC to start Tools Customizer. The first time that you use Tools Customizer, you must modify the settings that Tools Customizer uses to customize DB2 Admin.

Starting Tools Customizer

Start Tools Customizer by running a REXX EXEC from the ISPF Command Shell panel.

Before you begin

Tools Customizer must be SMP/E installed. You must know the high-level qualifier of where the Tools Customizer libraries reside. The high-level qualifier is considered to be all the segments of the data set name except the lowest-level qualifier, which is SCCQEXEC.

Attention: Ensure that Tools Customizer load libraries are not APF authorized. APF authorizing Tools Customizer libraries results in an abend.

About this task

To run the REXX EXEC, you must either change the placeholder in the EXEC for the high-level qualifier of the Tools Customizer EXEC library or pass the high-level qualifier as a parameter when you run the EXEC. The REXX EXEC is in the CCQTCZ member of the EXEC library.

Procedure

1. Optional: Change the placeholder for the high-level qualifier in the REXX EXEC:

    a. Find the EXEC library data set for Tools Customizer. The name of the data set is *high_level_qualifier*.SCCQEXEC.

    b. Edit data set member CCQTCZ and replace the `<TCZ HLQ>` string with the high-level qualifier of the EXEC library data set. For example, if the name of the Tools Customizer EXEC library is CCQTCZ.USABSAND.SCCQEXEC, replace `<TCZ HLQ>` with `CCQTCZ.USABSAND`.

    You have to change the placeholder for the high-level qualifier only once. When you run the REXX EXEC, you do not have to pass the high-level qualifier as a parameter.

2. Run the REXX EXEC (CCQTCZ):

    a. From the ISPF Primary Option Menu, select option 6. The ISPF Command Shell panel is displayed.

    b. Specify the EX command to run the REXX EXEC. For example, if the Tools Customizer EXEC library is CCQTCZ.USABSAND.SCCQEXEC and you changed the placeholder for the high-level qualifier in the REXX EXEC, specify: `EX 'CCQTCZ.USABSAND.SCCQEXEC(CCQTCZ)'`

    If you did not change the placeholder for the high-level qualifier in the REXX EXEC, specify: `EX 'CCQTCZ.USABSAND.SCCQEXEC(CCQTCZ)'` `'CCQTCZ.USABSAND'`

Results

The IBM Customizer Tools for z/OS main menu panel is displayed.

What to do next

If you are running Tools Customizer for the first time, you must modify the Tools Customizer user settings. If you have already set the Tools Customizer user settings, either customize or recustomize DB2 Admin.

Modifying Tools Customizer user settings

Before you can customize DB2 Admin with Tools Customizer, you must review the settings that Tools Customizer uses. You might have to change the default values to suit your environment. In most cases, you can change the Tools Customizer values at any time. For example, after you have customized DB2 Admin and are customizing a different product or solution pack, you might have to change the settings.

Procedure

1. On the IBM Tools Customizer for z/OS main panel (CCQPHME), specify option 0, **User settings for Tools Customizer**. The Tools Customizer Settings panel (CCQPSET) is displayed, as shown in the following figure:

```
CCQPSET                Tools Customizer Settings                14:03:51
Command ===>
Enter the settings for customizing a product or press End to save and exit.


Commands: SAVE - Save user settings


Product Customization Settings
  Customization library qualifier . . DB2TOOL.PRODUCT.CUST
  Use DB2 group attach  . . . . . . . YES (YES/NO)

Tools Customizer Library Settings
  Metadata library . . . . . DB2TOOL.CCQ110.SCCQDENU
  Discover output data set . DB2TOOL.CCQ110.DISCOVER
  Data store data set  . . . DB2TOOL.CCQ110.DATASTOR

User Job Card Settings for Customization Jobs
  ===> //        JOB
  ===>
  ===>
  ===>
  ===>
```

Figure 1. The Tools Customizer Settings panel (CCQPSET)

2. Review the values for the following required fields. Use the default value or specify your own value. You must have appropriate read and write access to the data sets that are specified.

Customization library qualifier

> The high-level qualifier that is used as the prefix for the customization library. The customization library is a data set in which the generated jobs to customize DB2 Admin are stored. Write access to this qualifier is required.

> For each product to be customized, the first value that is specified for the qualifier is always used, even if you change it after you have generated the customization jobs. For example, if you customize a product and then specify a new qualifier for recustomization, although the new qualifier is saved and displayed, the original value is used.

> To maintain multiple instances of Tools Customizer, specify a unique customization library qualifier for each instance of Tools Customizer. Data set names that exceed 42 characters must be enclosed in single quotation marks (').

Use DB2 group attach

> Determines the value that is used in the CONNECT statements in the generated customization jobs. Specify YES for data sharing environments, which causes the group attach name to be used. Specifying NO, in most cases, causes the SSID to be used in the DB2 CONNECT statement.

> **Important:** This field has no effect when you are customizing a product on a DB2 subsystem that is not a member of a data sharing group. In this case, the DB2 subsystem ID (SSID) is always used in the CONNECT statements in the generated customization jobs.

> When you are customizing a product on a DB2 subsystem that is a member of a data sharing group, how the DB2 subsystem is defined and the value of the **Use DB2 group attach** field determines the value that is used in the CONNECT statements in the generated jobs. The following table shows whether the SSID or the group attach name is used:

Table 4. The effect of the value of the **Use DB2 group attach** *field in a data sharing environment*

DB2 subsystem definition	Value of the Use DB2 group attach field	Value that is used in the CONNECT statements
The DB2 subsystem is defined with an SSID.	Yes	Group attach name
	No	SSID[1]
The DB2 subsystem is not defined with an SSID.	Yes or No	Group attach name

Note 1: If you generate jobs for multiple DB2 subsystems that are defined with an SSID and belong to the same data sharing group, the SSID of the first DB2 subsystem that is selected is used.

For example, assume that on the Customizer Workplace panel, you generated jobs for the following DB2 subsystems:

- V91C, which is a stand-alone DB2 subsystem
- V91A, which is a DB2 subsystem that is a member of data sharing group DSG1
- A DB2 subsystem that was not defined with an SSID that is a member of data sharing group DSGA

The following figure shows how these DB2 entries might be listed on the Customizer Workplace panel:

```
Associated DB2 Entries and Parameter Status
  Line commands: G - Generate jobs  E - Edit  B - Browse  C - Copy  R - Remove
  Cmd SSID GrpAttch Lvl Mode User ID Date       Status           Message
      V91C --       910 NFM  SYSADM  2010/11/09 Ready to Customize
      V91A DSG1     910 NFM  SYSADM  2010/11/09 Ready to Customize
      --   DSGA     910 NFM  SYSADM  2010/11/09 Ready to Customize
---------------------------- End of DB2 entries ----------------------------
```

The following table shows which values are used in the CONNECT statements in the generated jobs, based on the value of the **Use DB2 group attach** field.

Table 5. Value that is used in the CONNECT statements in the generated jobs

SSID	GrpAttch	Value of the Use DB2 group attach field	Value that is used in the CONNECT statements
V91C	--	Yes	SSID
		No	SSID
V91A	DSG1	Yes	Group attach name
		No	SSID
--	DSGA	Yes	Group attach name
		No	Group attach name

Tools Customizer metadata library

The name of the data set that contains the metadata that is used to display the DB2 and LPAR parameters. The parameters that are displayed on the LPAR Parameters panel and the DB2 Parameters panel depend on the parameters that you define and the tasks and steps that you select on the Product Parameters panel for the product that you are customizing. For example, the DB2 parameters that are required, based

on the selected tasks and steps, are displayed on the DB2 Parameters panel, and you can edit them. If they are not required, they are not displayed. Read access to this data set is required. Data set names that exceed 42 characters must be enclosed in single quotation marks (').

Discover output data set
> The name of the data set in which the output from the DB2 Admin Discover EXEC is stored. Each product has its own Discover EXEC. The Discover EXEC retrieves the product, LPAR, and DB2 parameters from a previously customized product. Write access to this data set is required. Data set names that exceed 42 characters must be enclosed in single quotation marks (').

Data store data set
> The name of the data set where Tools Customizer stores information about product, LPAR, and DB2 parameter values. Information about which products are associated with which DB2 entries (DB2 subsystems, DB2 group attach names, and DB2 data sharing members) is also stored in this data set. Data set names that exceed 42 characters must be enclosed in single quotation marks ('). The specified data store data set can be used with only one invocation of Tools Customizer at a time. Data set names that exceed 42 characters must be enclosed in single quotation marks (').

User job card settings for customization jobs
> The job card information to be inserted into the generated jobs for customizing a product. The default value is the job statement information from the ISPF Batch Selection panel.
>
> The first line of the job card automatically begins with the following information:
>
> ```
> //        JOB
> ```
>
> where characters 3 - 10 are reserved by Tools Customizer for the job name and includes a blank space after JOB. This name cannot be edited. Information that you specify on the first line of the job card cannot exceed 57 characters. This character limit includes a continuation character. All other lines of the job card cannot exceed 72 characters.

3. Press End to save and exit. If the Discover output data set and the data store data set that you specified do not exist, Tools Customizer creates them.

   Important: If the ISPF sessions unexpectedly ends before you exit Tools Customizer, the fields on the Tools Customizer Settings panel (CCQPSET) will be repopulated with default values, and you will be required to review them or specify new values again.

Results

The values are saved, and the IBM Tools Customizer for z/OS main menu panel (CCQPHME) is displayed again.

What to do next

You are ready to customize or recustomize DB2 Admin or to change parameter settings.

Related tasks:

Specifying the metadata library for the product to customize

You must specify a metadata library for the product that you want to customize.

About this task

The product metadata library contains the information that determines which tasks, steps, and parameters are required to customize DB2 Admin. This information controls what is displayed on the Product Parameters panel, the LPAR Parameters panel, and the DB2 Parameters panel.

After DB2 Admin has been SMP/E installed, the default name of the product metadata library is *high_level_qualifier*.SADBDENU, where *high_level_qualifier* is all of the segments of the data set name except the lowest-level qualifier.

Procedure

1. Specify option 1 on the Tools Customizer for z/OS panel. The Specify the Metadata Library panel is displayed. This panel contains a list of the product metadata libraries that you specified most recently. If you are using Tools Customizer for the first time, this list is empty, as shown in the following figure:

```
CCQPHLQ                   Specify the Metadata Library               14:50:11
Command ===>                                               Scroll ===> PAGE

Type the name of the metadata library for the pack or the product in the
Metadata library field, or select the library in the list of previous
libraries and press Enter to populate the field. Press Enter to continue.

The default name of the metadata library after the pack or product has been
SMP/E installed is <hlq>.SxxxDENU, where <hlq> is the high-level qualifier for
the pack or the product, and xxx is the 3-character prefix for the pack or
the product.

Metadata library . ADB.ADBB2PAR.SADBDENU

Previously Used Metadata Library:

=>
=>
=>
=>
```

Figure 2. The Specify the Metadata Library panel

2. Use one of the following methods to specify the product metadata library:
   - Type the name of a fully qualified partitioned data set (PDS) or an extended partitioned data set (PDSE) in the **Metadata library** field. Double quotation marks (") cannot be used around the name. Single quotation marks (') can be used but are not required. If you are customizing DB2 Admin for the first time, you must use this method.
   - Place the cursor on the library name in the Recent Metadata Libraries list, and press Enter.

Results

If you are customizing DB2 Admin for the first time, the Run Discover EXEC panel is displayed. Otherwise, the Customizer Workplace panel is displayed.

What to do next

- Complete the steps that correspond to your environment:

    Customizing DB2 Admin for the first time
    Do not run the DB2 Admin Discover EXEC. Press End. The Customizer Workplace panel is displayed. If your environment requires associated DB2 entries, ensure that they are created and associated. If your environment does not require associated DB2 entries, skip this step, and edit DB2 Admin parameters.

    Customizing DB2 Admin from a previous or current customization
    Press Enter to run the DB2 Admin Discover EXEC. The Discover Customized Product Information panel is displayed. Specify the required information for running the EXEC.

Discovering DB2 Admin information automatically

You can use the DB2 Admin Discover EXEC to discover information from a previous or current customization of DB2 Admin.

About this task

Tip: Using the DB2 Admin Discover EXEC to discover information from a previous or current customization saves time and reduces errors that can occur when parameters are specified manually.

DB2 Admin provides the Discover EXEC that you will run. Therefore, the information that can be discovered depends on DB2 Admin.

Parameter values that are discovered and parameter values that are specified manually are saved in the data store. If parameter values for the product that you want to customize exist in the data store, Tools Customizer issues a warning before existing values are replaced.

Procedure

1. On the Customizer Workplace panel, issue the `DISCOVER` command. If you chose to run the DB2 Admin Discover EXEC on the pop-up panel after you specified the product to customize, skip this step.

    Tip: You can run any Tools Customizer primary command by using either of the following methods:
    - Place the cursor on the name of the primary command, and press Enter.
    - Type the primary command name in the command line, and press Enter.

    The Discover Customized Product Information panel is displayed, as shown in the following figure:

```
CCQPDSC              Discover Customized Product Information           15:07:28
Command ===>                                              Scroll ===> CSR

For the product you are customizing, the Discover EXEC retrieves product
information from an already customized product. Specify the required
information. To save your information and run the Discover EXEC, issue the RUN
command. To save your information and stay on this panel, issue the SAVE
command. To verify the syntax of your information without saving it, press
Enter. To save and exit, press  End.

Commands: RUN  SAVE

Discover EXEC for Extracting Information from an Already Customized Product
 Discover EXEC library  . . . ADB.ADBB2PAR.SADBEXEC
 Discover EXEC name . . . . : ADB2CUST
 Discover output data set . . CCQTCZ.SYSADM.DISCOVER

Information for Discover EXEC
 Source Customized table library . . . . . .                              >
Target Customized table library . . . . . .                              >
DB2 Group Attach Name . . . . . . . . . . . NONE
Trace . . . . . . . . . . . . . . . . . .     (YES, NO)
```

Figure 3. The Discover Customized Product Information panel

2. Either accept the default values for the following input fields that Tools
   Customizer generates, or replace the default values with your own values:

   Discover EXEC library
   > The fully qualified data set name that contains the DB2 Admin
   > Discover EXEC.

   Discover EXEC name
   > The name of the DB2 Admin Discover EXEC.

   Discover output data set
   > The fully qualified data set where output from the DB2 Admin
   > Discover EXEC is stored.

3. Either accept or change the default values in the **Information for Discover
   EXEC** fields. These fields are generated by DB2 Admin. They show the
   information that is required to run the DB2 Admin Discover EXEC.

4. Issue the RUN command to run the DB2 Admin Discover EXEC. Alternatively,
   save your information without running the DB2 Admin Discover EXEC by
   issuing the SAVE command. If you issue the RUN command to run the DB2
   Admin Discover EXEC, the parameter information is discovered for DB2
   Admin, and the Customizer Workplace panel is displayed.

Results

The discovered parameter values for DB2 Admin replace any existing values.

What to do next

The next step depends on your environment:
- If DB2 entries were not discovered, or if you need to customize DB2 Admin on
  new DB2 entries, create and associate the entries.
- If DB2 entries were discovered and you want to customize DB2 Admin on only
  these entries, define the parameters.

Related tasks:

"Creating and associating DB2 entries" on page 84
You can create new DB2 entries and associate them with DB2 Admin.

"Defining parameters" on page 85
To customize DB2 Admin, you must define DB2 Admin parameters, LPAR parameters, and DB2 parameters, if your customization requires DB2 entries.

Creating and associating DB2 entries

You can create new DB2 entries and associate them with DB2 Admin.

About this task

The list of associated DB2 entries is on the Customizer Workplace panel.

Procedure

1. Issue the ASSOCIATE command on the Customizer Workplace panel. The Associate DB2 Entry for Product panel is displayed, as shown in the following figure:

```
CCQPDAD                   Associate DB2 Entry for Product        Row 1 to 3 of 3
Command ===>                                                     Scroll ===> CSR

Select any of the following DB2 entries to add them to the Customizer
Workplace panel. You use the Customizer Workplace panel to choose the DB2
subsystems, data sharing members, and group attach names on which to
customize the product.

Commands:  CREATE - Create a new DB2 entry

DB2 Entries
     Line commands: A - Associate  C - Copy  D - Delete
     Cmd SSID Grpattch
         DBAA  --
         DBAB  --
         DBAC  --

---------------------------- End of DB2 entries ----------------------------
```

Figure 4. The Associate DB2 Entry for Product panel

2. Create DB2 entries. If you need to associate DB2 entries that are already in the master list, skip this step and go to step 3.

   a. Issue the CREATE command. The Create DB2 Entries panel is displayed, as shown in the following figure:

```
CCQPCDB         Create DB2 Entries
 Command ===>

 Specify the SSID, the group attach name, or both in the appropriate columns
 for each new DB2 entry and press Enter. To create additional entries, issue
 the Inn line command, where nn is the number of entries to be inserted, and
 press Enter. To cancel, press End.

 New DB2 Entry Information
  Line commands: I - Insert into list   R - Remove from list
  Cmd  SSID  GrpAttch  Message
```

Figure 5. Create DB2 Entries panel

   b. In the appropriate columns, specify a DB2 subsystem ID, DB2 group attach name, or DB2 data sharing member name for the DB2 entry that you want to create, and press Enter. Valid values are 1 - 4 characters. You can use symbolic characters. You cannot use blanks.

Tips:

- To insert multiple DB2 entries, specify the I*nn* line command, where *nn* is the number of DB2 entries to be inserted.
- You will define specific parameters for these new DB2 entries, such as parameters that define a subsystem as primary, on the DB2 Parameters panel. This panel is displayed after you select these new DB2 entries and issue the line command to generate the jobs, after you issue the primary command to generate the jobs for all associated DB2 entries, or when you manually edit the DB2 parameters.

The Associate DB2 Entry for Product panel is displayed, and the new DB2 entry is displayed in the master list, as shown in the following figure:

```
CCQPDAD                   Associate DB2 Entry for Product         Row 1 to 3 of 3
Command ===>                                                      Scroll ===> CSR

Select any of the following DB2 entries to add them to the Customizer
Workplace panel. You use the Customizer Workplace panel to choose the DB2
subsystems, data sharing members, and group attach names on which to
customize the product.

Commands:  CREATE - Create a new DB2 entry

DB2 Entries
     Line commands: A - Associate  C - Copy  D - Delete
     Cmd SSID Grpattch
         DBAA  --
         DBAB  --
         DBAC  --
---------------------------- End of DB2 entries ----------------------------
```

Figure 6. Associate DB2 Entry for Product panel

  c. Repeat steps b and c for each DB2 entry that you want to create.
  d. When you have created all the DB2 entries, associate them with DB2 Admin, or press End to display the Customizer Workplace panel.

3. Associate DB2 entries.

  a. Specify A against one or more DB2 entries in the master list, and press Enter to associate them with DB2 Admin.

Results

The Customizer Workplace panel is displayed with the associated DB2 entries displayed in the associated list.

What to do next

Define the parameters.

Related concepts:

"Tools Customizer terminology" on page 963
Tools Customizer uses several unique terms that you should be familiar with before you begin to use Tools Customizer.

Defining parameters

To customize DB2 Admin, you must define DB2 Admin parameters, LPAR parameters, and DB2 parameters, if your customization requires DB2 entries.

Defining DB2 Admin parameters

DB2 Admin parameters are specific to DB2 Admin.

About this task

If you ran the DB2 Admin Discover EXEC, you must review the parameters that were discovered.

Procedure

1. Specify E next to the **Product parameters** field on the Customizer Workplace panel, and press Enter. The Product Parameters panel is displayed, as shown in the following figure. If other steps must be completed in a specific sequence before you define the DB2 Admin parameters, a note labeled **Important** will display the correct sequence on this panel.

```
CCQPPRD                    Product Parameters                     12:18:01
Command ===>                                           Scroll ===> PAGE

 Complete the following tasks to customize the products. The required tasks and
 steps are preselected. Ensure that all parameters are specified for each
 selected step within a task. Press End to save and exit.

 Commands: SAVE   VERIFYOFF
 Line Commands: / - Select

 Product customization library : XYZ.PRODUCT.CUST.$MVS1$.XYZ410
                                                              More:     +

  Usage Notes:
  Values specified here apply to every DB2 SSID unless you specify values
  for parameters with the same names on the DB2 Parameters panel. In this
  case, the values on the DB2 Parameters panel override the values on the
  Product Parameters panel.

 * General Customization
     * Customize
     *Admin Tool/OC CLIST  . DMTOOL.ADBB2PAR.SADBCLST
     *Admin Tool DBRM  . . . DMTOOL.ADBB2PAR.SADBDBRM
     *Admin Tool/OC EXEC . . DMTOOL.ADBB2PAR.SADBEXEC
     *Admin Tool/OC Load . . DMTOOL.ADBB2PAR.SADBLLIB
     *Admin Tool/OC Message  DMTOOL.ADBB2PAR.SADBMLIB
     *Admin Tool/OC Panel  . DMTOOL.ADBB2PAR.SADBPLIB
     *Admin Tool/OC Skeleton DMTOOL.ADBB2PAR.SADBSLIB
     *Admin Tool/OC Table . DMTOOL.ADBB2PAR.SADBTLIB
     *Customized Table lib . XYZ.DEVCUST.ISPTLIB
      Admin Tool hlq . . . . . . . . . .. . . . DMTOOL.ADBB2PAR
      CM Batch PROCLIB . . .
      SYSAFF for DB2 utilities . . . . . . . .
      WLM Environment Name . . . . . . . . . .
      System identification method . . . . . . JESID
     *Type of DB2 security exit  . . . . . . . STD
```

Figure 7. The Product Parameters panel

2. Select any required tasks and steps, and specify values for any parameters. After you select a task or step with a slash (/), put the cursor in the selected field and press Enter. If tasks, steps, and parameters are required, they are preselected with a slash (/). Otherwise, they are not preselected.

   All of the required parameters have default values, which you can either accept or change.

   Tips:

- For a detailed description of all input fields, put the cursor in the field, and press F1 or the key that is mapped to Help.
- The following elements apply to specific fields:
  - **Add...** is displayed when parameters can have multiple values but currently have only one value. To specify multiple values in these fields, place the cursor on **Add...**, and press Enter. Use the displayed panel to add or delete additional values.
  - **List...** is displayed when the complete list of valid values for the fields is too long to be displayed on the panel. To see the complete list of values, place the cursor on **List...**, and press F1 or the key that is mapped to Help.
  - **More...** is displayed when input fields contains multiple values. To see all of the values in the field, place the cursor on **More...**, and press Enter.
3. Optional: Select other tasks and steps with a slash (/) and press Enter to activate the input fields. Either accept or change the default values that are displayed.
4. Press End to save your changes and exit, or issue the SAVE command to save your changes and stay on the Product Parameters panel.

Results

The Customizer Workplace panel is displayed, and the status of the product parameters is Ready to Customize.

What to do next

If the status of other parameters on the Customizer Workplace panel is Incomplete or Discovered, edit these parameters.

Related tasks:

"Defining LPAR parameters"
LPAR parameters are parameters on the local LPAR that are required to customize DB2 Admin.

"Defining DB2 parameters" on page 89
DB2 parameters are parameters for a DB2 entry.

Defining LPAR parameters

LPAR parameters are parameters on the local LPAR that are required to customize DB2 Admin.

Procedure

1. Specify E next to the **LPAR parameters** field, and press Enter. The LPAR Parameters panel is displayed, as shown in the following figure:

```
CCQPLPR                        LPAR Parameters                        12:23:37
Command ===>                                               Scroll ===> PAGE

Browse the values for all of the LPAR parameters. Press End to exit.

 Product to Customize
   Product metadata library . : ADB.QADEVB.SADBDENU    >  LPAR  . . : 3090
   Product name . . . . . . . : DB2 Administration Too >  Version . : 11.2.0




 ISPF Libraries
  Message library  . . . . . . . . . . . . :                 >    Add...
  Panel library  . . . . . . . . . . . . . :                 >    Add...
  Skeleton library . . . . . . . . . . . . :                 >    Add...
  Table library  . . . . . . . . . . . . . :                 >    Add...
  Link list library  . . . . . . . . . . . :                 >    Add...
  ISPF Llib1 . . . . . . . . . . . . . . . :                        >
  ISPF Llib2 . . . . . . . . . . . . . . . :                        >

 Other Parameters
  Unit name for TSO work data sets . . . . : WDS
  Unit name for batch work data sets . . . : WDS
  Unicode translation technique  . . . . . : UTF-8
```

Figure 8. The LPAR Parameters panel

2. Specify values for all required parameters that are displayed. Many parameters
   have default values, which you can either accept or change.

   Tips:
   - For a detailed description of all input fields, put the cursor in the field, and
     press F1 or the key that is mapped to Help.
   - The following elements apply to specific fields:
     - **Add...** is displayed when parameters can have multiple values but
       currently have only one value. To specify multiple values in these fields,
       place the cursor on **Add...**, and press Enter. Use the displayed panel to
       add or delete additional values.
     - **List...** is displayed when the complete list of valid values for the fields is
       too long to be displayed on the panel. To see the complete list of values,
       place the cursor on **List...**, and press F1 or the key that is mapped to Help.
     - **More...** is displayed when input fields contains multiple values. To see all
       of the values in the field, place the cursor on **More...**, and press Enter.

   The following LPAR parameters can contain 1 - 64 values:
   - LPAR macro library
   - Message library
   - Panel library
   - Skeleton library
   - ISPF table input library
   - ISPF user profile library
   - File tailoring output library
   - Link list library
   - Command procedures library
   - Macro library
   - Link-edit library
   - Load library
   - Started task library name

3. Press End to save your changes and exit, or issue the SAVE command to save your changes and stay on the same panel.

Results

The Customizer Workplace panel is displayed, and the status of the LPAR parameters is Ready to Customize.

What to do next

If the status of other parameters on the Customizer Workplace panel is Incomplete or Discovered, edit these parameters.

Related tasks:

"Defining DB2 Admin parameters" on page 86
DB2 Admin parameters are specific to DB2 Admin.

"Defining DB2 parameters"
DB2 parameters are parameters for a DB2 entry.

Defining DB2 parameters

DB2 parameters are parameters for a DB2 entry.

About this task

If you did not run the DB2 Admin Discover EXEC, you must create and associate one or more DB2 entries before you can define the DB2 parameters. For more information, see "Creating and associating DB2 entries" on page 84.

Procedure

1. Specify E next to one or more DB2 entries in the associated list, which is in the Associated DB2 Entries and Parameter Status section on the Customizer Workplace panel, and press Enter. The DB2 Parameters panel is displayed, as shown in the following figure:

```
CCQPDB2                      DB2 Parameters: DB2 Admin Tool              13:36:01

 Ensure that values are specified for the required DB2 parameters. Press End
 to save and exit.

 Commands: SAVE  VERIFYOFF

  DB2 subsystem ID  . . . . . . . . . . . . : DB01
  Group attach name . . . . . . . . . . . . :
  Started task name for MSTR services . . . .

  General DB2 Information
   Mode . . . . . . . . . . . . . . . . . . NFM    (CM,CM8,CM9,NFM)
   Level Number . . . . . . . . . . . . . . 101 (101, 111)

  DB2 Utilities - common
   SYSAFF for DB2 utilities . . . . . . . . SY4A
   *Plan name for the DSNTEP2 utility  . . . . DSNTEP2
```

Figure 9. The DB2 Parameters panel

```
CCQPDB2                         DB2 Parameters                        11:18:05
 Command ===>                                            Scroll ===> PAGE

 Ensure that values are specified for the required DB2 parameters. Press End
 to save and exit.

 Commands: SAVE  VERIFYOFF
   DB2 subsystem ID . . . . . . . . . . . . : DSNB
   Group attach name . . . . . . . . . . . .
   Started task name for MSTR services . . . .

 General DB2 Information - common
    *Mode . . . . . . . . . . . . . . . . . . NFM (CM, CM8, CM9, NFM)
    *Level number . . . . . . . . . . . . . . 111 (101, 111)

 DB2 Utilities - common
    SYSAFF for DB2 utilities . . . . . . . . . SY4A
    *Plan name for the DSNTEP2 utility  . . . . DSNTEP2
```

Figure 10.

2. Specify values for all parameters that are displayed.

   Tips:
   - For a detailed description of all input fields, put the cursor in the field, and press F1 or the key that is mapped to Help.
   - The following elements apply to specific fields:
     – **Add...** is displayed when parameters can have multiple values but currently have only one value. To specify multiple values in these fields, place the cursor on **Add...**, and press Enter. Use the displayed panel to add or delete additional values.
     – **List...** is displayed when the complete list of valid values for the fields is too long to be displayed on the panel. To see the complete list of values, place the cursor on **List...**, and press F1 or the key that is mapped to Help.
     – **More...** is displayed when input fields contains multiple values. To see all of the values in the field, place the cursor on **More...**, and press Enter.

   Many parameters have default values, which you can either accept or change.

3. Press End to save your changes and exit, or issue the SAVE command to save your changes and stay on the same panel.

Results

The status of the DB2 entries that you selected on the Customizer Workplace panel is Ready to Customize.

What to do next

If the status of other parameters on the Customizer Workplace panel is Incomplete or Discovered, edit these parameters.

Related tasks:

"Defining DB2 Admin parameters" on page 86
DB2 Admin parameters are specific to DB2 Admin.

"Defining LPAR parameters" on page 87
LPAR parameters are parameters on the local LPAR that are required to customize DB2 Admin.

Generating customization jobs

To generate customization jobs for DB2 Admin and any associated DB2 entries, issue the `GENERATEALL` command, or select one or more DB2 entries on which to customize DB2 Admin.

Procedure

Generate the customization jobs by using one of the following methods.

- If you want to generate customization jobs at the product level and for any associated DB2 entries, issue the `GENERATEALL` command, and press Enter.
- If you want to generate customization jobs for specific DB2 entries, select the DB2 entries by specifying the `G` line command against them, and press Enter. The available DB2 entries are in the associated list in the Associated DB2 Entries and Parameter Status section.

Important: Regenerating customization jobs will replace any existing jobs, including jobs that you might have manually modified after they were generated.

Results

If the status is Incomplete or Discovered for DB2 Admin parameters, LPAR parameters, or DB2 parameters, Tools Customizer automatically starts an editing session for the types of parameters that are required. The session continues until the panel for each type of required parameter has been displayed.

What to do next

If an automatic editing session is started, accept the displayed parameter values or define values for the required types of parameters, select optional parameters, tasks, or steps for your environment, and save the parameter values. Otherwise, the customization jobs are generated, and you can submit them.

Tip: If the customization jobs are generated, but you are not ready to submit them, you can see them later by issuing the JOBLIST command on the Customizer Workplace panel. The JOBLIST command displays the Finish Product Customization panel, which you can use to submit the jobs.

Submitting customization jobs

Submit the customization jobs to customize DB2 Admin.

Before you begin

Ensure that the correct jobs are generated.

About this task

The following figure shows part of the Finish Product Customization panel. The table on this panel shows the customization jobs that are generated by Tools Customizer. They are grouped by job sequence number.

```
CCQPCST                    Finish Product Customization        Row 1 to 6 of 6
Command ===>                                                   Scroll ===> PAGE

  For a first-time customization, submit the jobs in the members in the order
  in which they apply to the DB2 entries. Otherwise, submit only the necessary
  jobs that were generated after changes were made. To submit jobs, browse
  the members and issue the TSO SUBMIT command.

  Line Commands: E - Edit  B - Browse

  Product customization library .: CCQTCZ.SYSADM.CUST.$3090$.ADB1120

Cmd Member   SSID GrpAttch Template Date       Description
-   -------- ---- ----     -------- ---------- ------------------------------->
    A0CUSTAA DSNB --       ADBCUST  2013/09/03 General customization
    A2SETUAA DSNB --       ADBSETUP 2015/10/07 Admin Tool Setup Task
    A6CMBAT  --   --       ADBCMBAT 2013/09/03 Create CM Batch JCL procedure
    A7CMBSAA DSNB --       ADBCMBSS 2013/09/03 Create CM Batch items
    A8BINDAA DSNB --       ADBBIND  2013/09/03 Binds
    B0CMBIAA DSNB --       ADBCMBIV 2015/10/07 Verify CM Batch JCL procedure
```

Figure 11. The Finish Product Customization panel

The member-naming conventions depend on whether the customization jobs are
for DB2 entries, and LPAR, or the product.

Customization jobs for DB2 entries

> The members use the following naming convention:
>
> *<job_sequence_number><job_ID><DB2_entry_ID>*
>
> where
>
> *job_sequence_number*
>> Two alphanumeric characters, A0 - Z9, that Tools Customizer
>> assigns to a customization job. The number for the first template in
>> the sequence is A0, the number for the second template is A1, and
>> so on.
>
> *job_ID*
>> Characters 4 - 7 of the template name, if the template name
>> contains five or more characters. Otherwise, only character 4 is
>> used. DB2 Admin assigns the template name.
>
> *DB2_entry_ID*
>> Two alphanumeric characters, AA - 99, that Tools Customizer
>> assigns to a DB2 entry.
>
> For example, the XYZBND*DB2_entry_ID_1* and XYZBND*DB2_entry_ID_2*
> jobs are generated from the XYZBNDGR template, and the
> XYZ4*DB2_entry_ID_1* and XYZ4*DB2_entry_ID_2* jobs are generated from
> the XYZ4 template. If the jobs are generated on two DB2 entries, the
> following member names are listed sequentially: A0BNDGAA,
> A0BNDGAB, A14AA, A14AB.

Customization jobs for an LPAR or the product

> The members use the following naming convention:
>
> *<job_sequence_number><job_ID>*
>
> where
>
> *job_sequence_number*
>> Two alphanumeric characters, A0 - Z9, that Tools Customizer

assigns to a customization job. The number for the first template in the sequence is A0, the number for the second template is A1, and so on.

job_ID

Characters 4 - 8 of the template name, if the template name contains five or more characters. Otherwise, only character 4 is used. For example, for the XYZMAKE template, the job ID is MAKE. For the XYZM template, the job ID is M. DB2 Admin assigns the template name, and it is displayed in the Template column.

For example, the XYZBNDGR job is generated from the XYZBNDGR template, and the XYZ4 job is generated from the XYZ4 template. The following member names are listed sequentially: A0BNDGR, A14.

Procedure

1. Submit the generated customization jobs by following the process that you use in your environment or by using the following method:
   a. Specify B against a customization job or the product customization library, and press Enter. An ISPF browsing session is started.
   b. Browse the customization job or each member in the library to ensure that the information is correct.
   c. Run the TSO SUBMIT command.
2. Press End.

Results

DB2 Admin is customized, and the Customizer Workplace panel is displayed. The status is Customized for the DB2 entries on which DB2 Admin was customized.

What to do next

You can generate more customization jobs for other DB2 entries, view a list of customization jobs that you previously generated, or recustomize DB2 Admin.

Browsing parameters

You can browse the product parameters, the LPAR parameters, and the DB2 parameters in read-only mode.

Procedure

1. On the Customizer Workplace panel, specify B next to the **Product parameters** field, the **LPAR parameters** field, or the DB2 entry that you want to browse, and press Enter. The panel that corresponds to your specification is displayed.
2. Press End to exit.

Copying DB2 entries

You can copy associated and not associated DB2 entries to other DB2 entries or to new DB2 entries.

About this task

Go to the step that applies to your environment:

- To copy an associated DB2 entry to another associated DB2 entry or to an entry that is not associated, go to step 1.
- To copy an associated DB2 entry to a new entry, go to step 2.
- To copy a DB2 entry that is not associated to a new entry, go to step 3.

Procedure

1. To copy an associated DB2 entry to another associated DB2 entry or to an entry that is not associated, complete the following steps:

   a. Specify C against a DB2 entry in the associated list of DB2 entries on the Customizer Workplace panel, and press Enter. The Copy Associated DB2 Entry panel is displayed.

   b. Select one or more DB2 entries to which information will be copied by specifying the / line command, and press Enter. The Associated column indicates whether the DB2 entry is associated.

   Tip: To copy information into all of the DB2 Entries in the list, issue the SELECTALL primary command, and press Enter.
   The Copy DB2 Parameter Values panel is displayed.

   c. Specify an option for copying common and product-specific DB2 parameter values. Common DB2 parameter values apply to all DB2 entries for all products that you have customized by using Tools Customizer. Product-specific DB2 parameter values apply only to the product that you are currently customizing.

   - To copy the common DB2 parameter values and the product-specific DB2 parameter values, specify option 1, and press Enter.
   - To copy only the product-specified DB2 parameter values, specify option 2, and press Enter.

   In some cases, the DB2 parameter values might contain the DB2 subsystem ID as an isolated qualifier in data set names. For example, in the DB01.DB01TEST.DB01.SANLLOAD, data set name, the DB01 subsystem ID is isolated in the first and third qualifiers but is not isolated in the second qualifier. When the DB2 subsystem ID is an isolated qualifier in data set names, the Change DB2 Subsystem ID in DB2 Parameter Values panel is displayed. Otherwise, the Customizer Workplace panel is displayed.

   d. If the Change DB2 Subsystem ID in DB2 Parameter Values panel is displayed, specify an option for changing the subsystem IDs. Otherwise, skip this step.

   - To change the subsystem ID in isolated qualifiers in data set names, specify option 1, and press Enter.
   - To use the same subsystem ID in all values, specify option 2, and press Enter.

   The Customizer Workplace panel is displayed with the copied associated entry in the list.

2. To copy an associated DB2 entry to a new entry, complete the following steps:

   a. Specify C against a DB2 entry in the associated list of DB2 entries on the Customizer Workplace panel, and press Enter. The Copy Associated DB2 Entry panel is displayed.

   b. Issue the CREATE command. The Create DB2 Entries panel is displayed.

   c. Specify the SSID, the group attach name, or both in the appropriate columns for each new DB2 entry, and press Enter.

> **Tip:** To add rows for additional entries, specify the I*nn* line command, where *nn* is the number of entries to be created, and press Enter. The Copy Associated DB2 Entry panel is displayed with the new entries in the list. The new entries are preselected.

    d. Press Enter to complete the copy process. The Customizer Workplace panel is displayed with the copied entries in the list.

3. To copy a DB2 entry that is not associated to a new entry, complete the following steps:

    a. Issue the ASSOCIATE command on the Customizer Workplace panel. The Associate DB2 Entry for Product panel is displayed.

    b. Select one or more DB2 entries by specifying the / line command, and press Enter. The Copy a DB2 Entry panel is displayed.

    c. Specify the SSID, the group attach name, or both in the appropriate columns for the new DB2 entry, and press Enter. The Associate DB2 Entry for product panel is displayed with the copied entry in the list.

    d. If you want to associate the copied entry, specify A against it, and press Enter. The Customizer Workplace panel is displayed with the copied entries in the list.

What to do next

Edit any of the parameters or generate the jobs.

Related concepts:

"Tools Customizer terminology" on page 963
Tools Customizer uses several unique terms that you should be familiar with before you begin to use Tools Customizer.

Removing DB2 entries

You can remove DB2 entries from the associated list.

About this task

When you remove DB2 entries from the associated list, any customization jobs for the entries are removed from the list of jobs on the Finish Product Customization panel, and they are deleted.

Procedure

On the Customizer Workplace panel, specify R next to one or more DB2 entries that you want to remove, and press Enter. The selected DB2 entries are removed from the associated list and added to the master list on the Associate DB2 Entry for Product panel, and the customization jobs are deleted.

Related concepts:

"Tools Customizer terminology" on page 963
Tools Customizer uses several unique terms that you should be familiar with before you begin to use Tools Customizer.

Deleting DB2 entries

You can delete DB2 entries from the master list.

About this task

When you delete DB2 entries from the master list, any associations and all customization jobs for products that are customized on the entries will be deleted.

Procedure

1. On the Customizer Workplace panel, issue the ASSOCIATE command. The Associate DB2 Entry for Product panel is displayed.
2. Specify D next to one or more DB2 entries that you want to delete, and press Enter. If the entry is associated with any products, the Delete Associated DB2 Entry panel for the first DB2 entry that you selected is displayed. Otherwise, the Delete DB2 Entry panel is displayed.
3. To delete the DB2 entries, press Enter. If the DB2 entries are associated with any products in the table on the Delete Associated DB2 Entry panel, any associations and all customization jobs for the products that are customized on it are deleted. Otherwise, only the DB2 entries are deleted. If you selected multiple DB2 entries to delete, the next DB2 entry that you selected is displayed on either the Delete Associated DB2 Entry panel or the Delete DB2 Entry panel. Otherwise, the Associate DB2 Entry for Product panel is displayed.

What to do next

If you selected multiple DB2 entries to delete, repeat step 3 until all selected entries are deleted. Then, continue the customization process.

Displaying customization jobs

You can view a list of the members that contain the customization jobs before or after you submit the jobs.

About this task

The customization jobs that you generate for one DB2 entry are also displayed when you customize DB2 Admin for another DB2 entry later.

Procedure

On the Customizer Workplace panel, issue the JOBLIST command. The Finish Product Customization panel is displayed. This panel shows the list of jobs that you have previously generated. They are grouped by job sequence number. Use this panel to browse or edit the generated jobs before you submit them.

Maintaining customization jobs

Instead of deleting customization jobs outside of Tools Customizer, you can maintain the correct jobs for DB2 Admin by completing the steps for recustomization.

About this task

You cannot delete or rename customization jobs from the customization library by starting an ISPF browse or edit session from the Finish Product Customization panel. If you try to delete customization jobs by using this method, the CCQC034S message is issued. If you try to rename customization jobs, the CCQC035S message is issued.

If you delete or rename customization jobs from the customization library by using ISPF outside of Tools Customizer, Tools Customizer will not recognize that the jobs were deleted, and the Finish Product Customization panel will still display them. If you browse or edit jobs that were deleted from the library outside of Tools Customizer, the CCQC027S message is issued.

Procedure

To maintain the correct customization jobs in the customization library, complete the steps for recustomization.

Using Tools Customizer in a multiple-LPAR environment

Currently, Tools Customizer supports only the local LPAR; however, you can propagate customizations to additional LPARs by using either of two different methods.

About this task

In a multiple-LPAR environment, Tools Customizer identifies the LPAR to which you are logged on. Tools Customizer uses this LPAR name for several different parameter settings, one of which is the data store. When you use the data store during the customization of DB2 Admin that is on a different LPAR, Tools Customizer issues message CCQD586S, which indicates that the product has already been customized based on values from the data store on the first LPAR. This message is issued to prevent the data store from becoming corrupted.

This behavior occurs in the following conditions:
- Tools Customizer is installed on a DASD device that is shared by multiple LPARs.
- After a product is customized by using Tools Customizer, the data store is copied to another LPAR.

Procedure

To customize products running against a DB2 subsystem on an LPAR where Tools Customizer is not installed, consider using one of the following methods:

Install one instance of Tools Customizer on one LPAR
    If you intend to reuse the customization values for all the instances of your products on all LPARs, use this method.
    1. Associate all the DB2 entries in this one instance of Tools Customizer. The LPARs on which the DB2 subsystems reside do not matter.
    2. Generate the customization jobs for each DB2 entry.
    3. Copy the generated customization jobs to the LPAR to run against the specific DB2 entries. Some LPAR-specific edits might be required. You can make these edits in the customized jobs that you copied. Note that this situation is one of the few situations where you might need to make manual changes to the jobs that are customized by Tools Customizer.

Install one instance of Tools Customizer on each LPAR
    If you do not want to reuse previous customization values and you want to start new customizations, use this method.

Important: This method will likely not be the preferred approach for most organizations because most organizations tend to use similar or identical customization values for each product instance on all LPARs.

Optional DB2 Admin customization tasks

Using Tools Customizer, you can make the following optional customizations.

Required in some cases: Update the APF Authorization table

You must update SYS1.PARMLIB to authorize the ADB2ATH and ADB2UTIL programs and the ADB2ATH and ADB2UTIL TSO commands.

Copy authorized programs ADB2ATH and ADB2UTIL from *high-level*.SADBLINK to an APF-authorized library OR an APF-authorized library in the system link list

Note: This APF-authorized library must either be in the system link list, or must be registered as the "DB2 Admin APF Library" on the Product Parameters panel (CCQPPRD).
The TSO service facility must invoke ADB2ATH and ADB2UTIL as authorized programs. Modify SYS1.PARMLIB(IKJTSO*xx*) and add programs ADB2ATH and ADB2UTIL, as shown in the following figure. Adding ADB2ATH and ADB2UTIL to SYS1.PARMLIB (IKJTSOxx) will allow the TSO service facility to invoke them as authorized.

The ADB2ATH program is used when the DB2 security exit type (:secexit.) is specified as AUTH. The ADB2UTIL program is used when ADBTEP2 runs DB2 utilities.

```
AUTHPGM NAMES(                    /* AUTHORIZED PROGRAMS */      +
  ........                                                       +
  ADB2ATH                    /* CALLS DSN3@ATH      */      +
  ADB2UTIL                   /* CALLS DSNUTILB      */      +
  ........)                  /*                     */      +
AUTHTSF NAMES(   /* PROGRAMS TO BE AUTHORIZED      */      +
                 /* WHEN CALLED THROUGH THE TSO    */      +
                 /* SERVICE FACILITY.              */      +
  ........                   /*                     */      +
  ADB2ATH                    /* CALLS DSN3@ATH      */      +
  ADB2UTIL                   /* CALLS DSNUTILB      */      +
  ........)                  /*                     */
```

Figure 12. Adding programs ADB2ATH and ADB2UTIL

Activate the changes immediately or at the next IPL by issuing the following TSO/E command:

```
PARMLIB UPDATE(xx)
```

Before using HPU within a work statement list, be sure to enable HPU. The main HPU program (INZUTILB) needs to be authorized in the IKJTSOnn member of PARMLIB.

Using two different versions of DB2 Admin on the same DB2 subsystem

If you are installing and running two or more different releases of DB2 Admin on the same DB2 subsystem, ensure that you use the ADB2ATH and ADB2UTIL authorized programs from the product tape for the higher release when you copy them to the APF-authorized library in your system link list.

Required in some cases: Specify a unicode translation technique parameter value

You might need to specify the technique for unicode translation.

The value in the **Unicode translation technique** field is derived from the CCSID conversion string, 01208. CCSID 01208 specifies the most recent UTF-8 version supported.

To find the value you need to specify:
1. Open a 3270 emulation session and find the 3270 emulation CCSID value, *xxx*, on the Session Parameters - 3270 Host panel, in the **HostCode-Page** field. In this example, the CCSID value is 037.
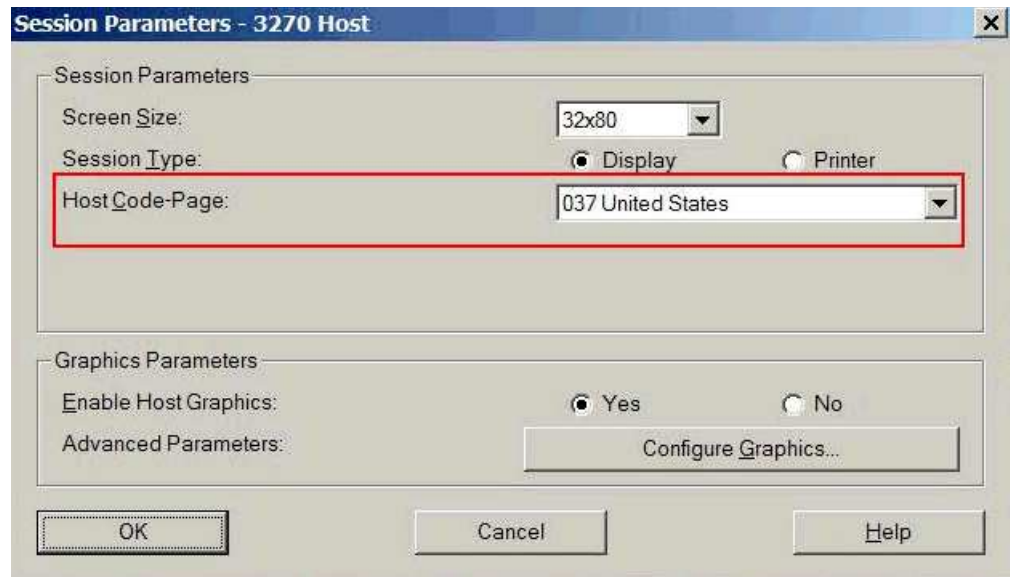
Figure 13. Session Parameters - 3270 Host panel

2. From the MVS™ log, run /display uni,all. Find `01208-xxxxx` (01208-00037 in this example) in the /D UNI,ALL output (at the bottom of this example). The suffix on the string 01208-00037 is the value you need to specify. In this example, the value is **E**.

```
   DISPLAY UNI,ALL
CUN3000I 12.11.38 UNI DISPLAY 216
 ENVIRONMENT: CREATED        10/01/2009 AT 07.41.34
              MODIFIED       10/01/2009 AT 07.48.28
              IMAGE CREATED --/--/---- AT --.--.--
     SERVICE: CHARACTER      CASE            NORMALIZATION  COLLATION
              STRINGPREP     BIDI
     STORAGE: ACTIVE         273 PAGES
              FIXED            0 PAGES
              LIMIT         1280 PAGES
    CASECONV: ENABLED
    CASE VER: UNI300  NORMAL
   NORMALIZE: DISABLED
    NORM VER: NONE
     COLLATE: DISABLED
  COLL RULES: NONE
 STRPROFILES: NONE
  CONVERSION: 00850-01200(13488)-R        01200(13488)-00037-E
              01200(13488)-00367-E        01047-01200(13488)-R
              01047-01200(13488)-L        01200(13488)-00500-E
      01047-01200(13488)-L        01200(13488)-00500-E
      01200(13488)-00819-E        01200(13488)-00850-E
      01208-00037-E               01200(13488)-01047-E
```

Figure 14. /D UNI,ALL output

3. Type the value, E, in the **Unicode translation technique** field.

Optional: Migrate modes

You can migrate from one release or mode of DB2 to another. For example, you can migrate from compatibility mode (CM) to new-function mode (NFM).

Procedure

1. Submit the ADBBIND job generated by Tools Customizer on all new DB2 subsystems. Submitting the ADBBIND job ensures that the changes made to the DB2 catalog are reflected in the product's behavior.

   Note: You can ignore BIND errors when running ADBBIND if the errors are related to the DB2 catalog tables.
2. Optional: If you have defined multiple copies of the DB2 catalog before upgrading to a new release or mode, re-run the bind steps for the catalog copies that you created.

Required in some cases: Tailor DB2 Admin Authorization Switching

DB2 Admin Authorization Switching is a facility within DB2 Admin that is used to execute DDL and DCL under the authority of another user. The facility does not cover other statement types, including DB2 Utility commands and DSN subcommands such as FREE PACKAGE and BIND PLAN.

About this task

This other user is called the *auth-switch ID*, and the ID that submits the job is called the *submitter*.

Alter Tablespace ALT, Alter Table ALT, WSLs, Change Management, Change Management batch, and DB2 Object Comparison Tool make use of authorization switching. These functions allow table spaces and tables to be redefined, which requires that they, and any dependent objects, be dropped and re-created.

However, the job submitter might not have the necessary authority to rebuild all the objects and authorizations. Authorization switching allows the job submitter to use an ID that does have the authority to run the DDL to rebuild the objects.

Before DB2 Admin Authorization Switching can be used, some additional installation steps must be performed to enable and protect it.

To complete the installation of DB2 Admin Authorization Switching:

Procedure

Create RACF profiles or equivalent (as required) to protect the facility.

Tip: When DB2 Admin Authorization Switching is enabled for a DB2 subsystem, create a RACF profile to protect the facility from unauthorized use. When DDL that is enabled with DB2 Admin Authorization Switching capability is run, a RACF access check is made to a resource that is intended to protect the use of a given Authorization Switching ID on the DB2 subsystem. The resource is within the IBM-supplied RACF FACILITY class in the following form:

```
ADBAUTHS.ssid.auth-switch-id
```

Example

If the DB2 subsystem is DSN and the desired authorization ID to use is SYSADMZ1, the RACF resource name that DB2 Admin generates is:

```
FACILITY ADBAUTHS.DSN.SYSADMZ1
```

For DB2 Admin Authorization Switching to proceed, the job submitter requires READ authority to the profile that protects this resource. The standard RACF profile rules apply for this resource. An installation can use general or more granular profile controls, as listed in the following table.

Table 6. Controlling the granularity of profiles.

Granularity	Example
A single profile that protects all subsystem/user ID combinations	FACILITY ADBAUTHS.*
A more granular profile	FACILITY ADBAUTHS.DSN.*
The finest degree of control	FACILITY ADBAUTHS.DSN.SYSADMZ1

If the FACILITY class is a RACLIST profile, the profiles must be refreshed after each change using the RACF SETROPTS command.

Restriction: DB2 Admin Authorization Switching requires that the RRS Attach Facility (RRSAF) of DB2 for OS/390® and z/OS is available.

Optional: Prepare ADBL CLIST

The ADBL CLIST in the SADBCLST library is provided for running DB2 Admin.

The ADBL CLIST brings up the DB2 Admin Main Menu.

You can invoke the ADBL CLIST from any ISPF panel or from the ISPF command processor panel (usually ISPF option 6). You can add the % prefix to the beginning of the CLIST name to ensure that TSO/E only searches the CLIST libraries.

Several CLIST parameters are available for your use:

ASUSER

Allows you to establish a trusted context. This parameter is passed to the DSN command. This parameter can also be used in ADB CLIST. If ASUSER is specified, then all additional connects made online should also be made using the specified ASUSER.

Restriction:

- ASUSER is only used in DSN connections.
- If the trusted context does not include the job name submitted, the trusted context will not be established.

CMD

An external product can invoke CLIST ADBL with an optional new keyword parameter, CMD, containing a catalog navigation command with an optional object type and an optional search criteria. The END command (PF3) returns you to the panel where the catalog navigation command was entered. When CMD is specified, the first token must be CAT.

Example:

```
%ADBL SYSTEM(DSNA) CMD('CAT T SYSTEM01%.TEST%')
```

Note: CMD is mutually exclusive with the **PANEL** and **DMT** parameters.

CMOWN

The owner (qualifier) of the Change Management database objects. If a minus sign value is used with this parameter, then a null value will be used instead of the value that was established during Tools Customizer install time.

The following examples show how you can use the CMOWN parameter:
```
CMOWN(CMDBADM)
CMOWN(-)
```

DASD

The unit name for batch work data sets. If you use a minus sign with this parameter, the value in the **Unit name for batch data sets** field on the LPAR Parameters panel is overridden by the DB2 Admin default, which is SYSDA.

The following examples show how you can use the DASD parameter:
```
DASD(SYSALLDA)
DASD(-)
```

DB2LLIB

List of the DB2 product load module libraries where DB2 is installed if DB2 is not in the linklist.

DEBUG

Use this parameter only at the request of your IBM service representative.

DMT

You can use the DMT parameter to access the DB2 Tools Launchpad panel. From this panel, you can either invoke a DB2 tool or you can continue to use the DB2 Admin functions that are described in this information. If you do not use the DMT parameter, you go directly to the DB2 Admin functions, but you cannot launch other DB2 tools from within DB2 Admin.

Restriction: If there is no active ISPF LIBDEF data set for table input library ISPTLIB, the DB2 Tools Launchpad can not be accessed correctly when using the DMT parameter.

DMTID

Indicates which library from the DB2 Tools Launchpad TLIB list you want to select by default. You can specify this parameter from your local front-end panel, CLIST, or from a REXX exec that invokes the ADBL CLIST. You will specify a number in parenthesis, for example, dmtid(2). An S will be placed in the Sel field for the row that you indicate (in this example, the second row) for the library that you want to be the default. This library is displayed in the panel to show where the update will be written. If the number you enter exceeds the number of rows, an S will be placed in the last row. If Launchpad is not active, then DMTID is ignored.

DUMP

Use this parameter only at the request of your IBM service representative.

INSTALL

Installation name.

JES

The JES environment name. For JES3 environments, specify JES(JES3). Otherwise, use the default (null).

LIBAPRE

The prefix for PRODADD() libraries. The default is none. See the LIBPRE parameter for an example of how data set names are generated from the LIBAPRE parameter.

LIBPRE

The prefix for DB2 Admin libraries. This prefix designates the first set of characters (up to four) in the final qualifier of the DB2 Admin libraries.

The default is SADB.

The following example shows how you can use the LIBPRE parameter to generate dataset names `ADB.SAMP.ISPPLIB` and `ADB.SAMP.ISPLLIB`:

```
PROD(ADB.SAMP) LIBPRE(ISP)
```

LIBDEF(YES|NO)

To access DB2 Admin, it is not necessary for the ADBL CLIST to issue ISPF LIBDEF statements for DB2 Admin libraries if you allocate those libraries in your TSO logon procedure. LIBDEF(YES) is the default. Specify LIBDEF(NO) to bypass the ISPF LIBDEFs.

Note: If you specify the parameter DMT with LIBDEF(NO) to access DB2 Tools Launchpad, you must ensure that there is already an active ISPF LIBDEF data set for the table input library ISPTLIB. If there is no active ISPF LIBDEF data set, you must first perform an ISPF LIBDEF statement for the library ISPTLIB, Otherwise, you can not access the DB2 Tools Launchpad if you allocate only the DB2 Admin libraries in your TSO logon procedure.

The following sample REXX EXEC performs the ISPF LIBDEF statement for the library ISPTLIB:

```
/* REXX */
/* Sample REXX EXEC LIBDEF */
Address ISPEXEC
"LIBDEF ISPTLIB DATASET ID('ADB.V720.SADBTLIB')"
exit
```

Example:

```
%LIBDEF
```

performs an ISPF LIBDEF statement for the following data set: ISPTLIB
DATASET ADB.V720.SADBTLIB.

To clear the above ISPF LIBDEF data set after setting it, you can perform the
following sample REXX EXEC:

```
/* REXX */
/* Sample REXX EXEC CRLIBDEF */
Address ISPEXEC
"LIBDEF ISPTLIB "
exit
```

LIST

High-level qualifiers of additional libraries to allocate before PROD(),
PRODADD(), and USERADD(). No default exists. If you specify LIST, you
must also specify LISTPRE. The entries that are specified in LIST and LISTPRE
have a one-to-one correspondence.

LISTPARM

Use this parameter, which causes a list of the initialization parameters to be
displayed, only at the request of your IBM service representative.

LISTPRE

List of prefixes for LIST() libraries. No default exists. If you specify LISTPRE,
you must also specify LIST. The entries that are specified in LIST and LISTPRE
have a one-to-one correspondence.

NEWAPPL

The ISPF application ID. NEWAPPL identifies the member name in which the
ISPF profile variables are saved for DB2 Admin. The default value for
NEWAPPL is null with an application ID of ISR. If you use a minus sign with
this parameter, the value set for the :newappl. tag is overridden by the DB2
Admin default, which is ISR.

The following examples show how you can use the NEWAPPL parameter:
```
NEWAPPL(ADB)
NEWAPPL(-)
```

PANEL

The panel name for the DB2 Admin panel that is displayed first. The default is
ADB2.

PGM

The name of the DB2 Admin main program. The default is ADBMAIN.

PLAN

The plan name to use. If you do not specify a plan name, the following plan
names are used: ADB, ADB2GEN, and ADB27AC. If you specify a plan name,
it is used for all programs.

PROD

You can use the PROD parameter to override the high-level qualifier for all
DB2 Admin product libraries, or you can edit the ADBL CLIST and specify the
high-level qualifier in the PROD parameter. On this parameter, you must
specify the correct value for the DB2 Admin libraries, including libraries that
are allocated in your TSO logon procedure. Specify a period to disable,
PROD(.). TSO does not allow PROD().

PRODADD

The high-level qualifier for additional product libraries to allocate in front of
PROD(). The default is none.

QTAB

Use this parameter, which lists open ISPF tables at the beginning and end of a DB2 Admin session, only at the request of your IBM service representative.

SECEXIT

The DB2 security exit type. The possible values are STD (the default), SAMPLE, AUTH, OWN, and NOCALL. If you use a minus sign with this parameter, the value set for the :secexit. tag or the **DB2 Security exit type** field on the Product Parameters panel is overridden by the DB2 Admin default, which is STD.

The following examples show how you can use the SECEXIT parameter:
```
SECEXIT(AUTH)
SECEXIT(-)
```

SHOW

Use the SHOW parameter to start your DB2 Admin session with a panel that shows all of the active DB2 subsystems that are available to you.

SYSTEM(*ssid*)

Use the SYSTEM(*ssid*) parameter to directly access a specific DB2 subsystem. This parameter is ignored if the SHOW parameter is specified.

USER

To activate the CLIST and EXEC libraries that are allocated to the SYSUPROC and SYSUEXEC DD names, issue an ALTLIB USER statement after ALTLIB APPLICATION. These libraries are then searched before searching the DB2 Admin libraries.

USERADD

The high-level qualifier for additional user-development libraries to allocate in front of PROD() and PRODADD(). The default is none.

USERPRE

The prefix for USERADD() libraries. The default is none. See the LIBPRE parameter for an example of how data set names are generated from the USERPRE parameter.

VB If your site uses variable-length CLIST and EXEC libraries, you can use the VB parameter to access the SADBCLST.VB and SADBEXEC.VB libraries that are created during installation.

VIO

The unit name for TSO work data sets. If you use a minus sign with this parameter, the value in the **Unit name for TSO work data sets** field on the LPAR Parameters panel is overridden by the DB2 Admin default, which is VIO.

The following examples show how you can use the VIO parameter:
```
VIO(SYSALLDA)
VIO(-)
```

When using the ADBL CLIST to define the DB2 Admin libraries (using the ISPF LIBDEF command), the data set naming convention must include the following components:
- A high-level qualifier specified using PROD, PRODADD, and USERADD.
- A prefix specified using LIBPRE, LIBAPRE, and USERPRE.
- A suffix that must include the following information:
  - LLIB load library
  - MLIB message library

- – PLIB panel library
- – SLIB skeleton library
- – TLIB table library
- – CLIST library CLIST or CLST
- – EXEC library

Three pairs of keyword parameters are used to specify up to three levels of concatenation for product libraries:

PROD and LIBPRE
> Used for the DB2 Admin libraries.

PRODADD and LIBAPRE
> Used for any additional products, such as DB2 Object Comparison Tool.

USERADD and USERPRE
> Used for user-private development libraries when adding your own panels to DB2 Admin.

The following example shows how you can use the three pairs of keyword parameters:

```
ADBL PROD(ADBB10) LIBPRE(SADB)
     PRODADD(GOCB10) LIBAPRE(SGOC)
     USERADD(USER01) USERPRE(ISP)
```

Using these parameter values results in allocating libraries as follows:

```
load library      USER01.ISPLLIB,GOCB10.SGOCLLIB, ADBB10.SADBLLIB
message library   USER01.ISPMLIB,GOCB10.SGOCMLIB, ADBB10.SADBMLIB
panel library     USER01.ISPPLIB,GOCB10.SGOCPLIB, ADBB10.SADBPLIB
skeleton library  USER01.ISPSLIB,GOCB10.SGOCSLIB, ADBB10.SADBSLIB
table library     USER01.ISPTLIB,GOCB10.SGOCTLIB,ADBB10.SADBTLIB
CLIST library     USER01.CLIST,GOCB10.SGOCLST,ADBB10.SADBCLST
exec library      USER01.EXEC,GOCB10.SGOCEXEC,ADBB10.SADBEXEC
```

In general, the last data set name qualifier is composed of the prefix (for example, SADB), followed by the library type (for example, LLIB). The exception to this convention is the prefix ISP, which generates data set names that use the qualifiers CLIST and EXEC for the CLIST and EXEC libraries, respectively.

The following sample REXX EXEC runs DB2 Admin:

```
/* REXX */
/* Sample REXX EXEC ADBRUN */
trace "O"
parse upper arg rel userparms
prod     = "PROD(.)"; libpre  = ""
adblclst = "'ADBB10.SADBCLST(ADBL)'"
plan     = "PLAN(ADB)"
select
  when rel = "PROD" then do
    list     = "LIST(''ADBA10'')"
    listpre  = "LISTPRE(''ISP'')"
    end
  when rel = "TEST" then do
    list     = "LIST(''USER.V10 ADBB10'')"
    listpre  = "LISTPRE(''SADB ISP'')"
    end
  otherwise do
    say "Invalid parameter:" rel "TEST assumed."
    list     = "LIST(''USER.V10 ADBB10'')"
    listpre  = "LISTPRE(''SADB ISP'')"
    end
```

```
        end /* select */
        say "CLIST  =" adblclst
        say "LIST   =" list listpre
        say "PARMS  =" plan userparms
        cmd = "EXEC" adblclst "'" prod list listpre plan userparms "'"
        say "CMD    =" cmd
        cmd
        exit
```

Example:

```
        %ADBRUN TEST
```

`will ISPF LIBDEF the following data sets:`

```
   ISPLLIB  DATASET      USER.V10.SADBLLIB
                         ADBB10.SADBLLIB

   ISPMLIB  DATASET      USER.V10.SADBMLIB
                         ADBB10.SADBMLIB

   ISPPLIB  DATASET      USER.V10.SADBPLIB
                         ADBB10.SADBPLIB

   ISPSLIB  DATASET      USER.V10.SADBSLIB
                         ADBB10.SADBSLIB

   ISPTLIB  DATASET      USER.V10.SADBTLIB
                         ADBB10.SADBTLIB
```

`and ALTLIB ACTIVATE the following data sets:`

```
   APPLICATION(CLIST)  DATASET       USER.V10.SADBCLST
                                     ADBB10.SADBCLST

   APPPLICATION(EXEC)  DATASET       USER.V10.SADBEXEC
                                     ADBB10.SADBEXEC
```

Optional: Verify activation of limited functionality

To use DB2 Admin you must check your TSO LOGON PROC and your link list definition to verify that the DB2 libraries are available to your TSO session.

About this task

After having submitted the BIND job (and the fixed to variable-block conversion job, if necessary), you can use DB2 Admin with limited functionality. However, to use DB2 Admin, the DB2 libraries must be allocated to your TSO session.

Before proceeding to enable DB2 Admin for additional functionality, verify that DB2 Admin is operational by completing the following steps:

Procedure

1. Determine where the DB2 DSN command is installed.
2. Make the DB2 Admin ISPF and TSO libraries available to your ISPF session. You can do this in one of the following ways:
   - A. Copy the ADBL CLIST to your standard ISPF and TSO libraries.
   - B. Add the DB2 Admin product libraries to your TSO LOGON PROC. The following table shows the libraries you can allocate to your TSO LOGON PROC.

Table 7. Libraries to allocate to your TSO LOGON PROC

DDNAME	Library DSN Suffix
ISPLLIB	SADBLLIB
ISPMLIB	SADBMLIB
ISPPLIB	SADBPLIB
ISPSLIB	SADBSLIB
ISPTLIB	SADBTLIB
SYSPROC	CLIST or SADBCLST
SYSPROC	EXEC or SADBEXEC

- C. Write a small CLIST that runs the ADBL CLIST.
3. Start DB2 Admin according to the option you chose in 2 on page 107.
    - If you chose option A or B, issue the following command:

      `TSO %ADBL`

    - If you chose option A or B and the DB2 DSN command is not in the linklist, you need to specify the data set name of the DB2 load module library in which the DSN command resides as a parameter when you issue the following command:

      `TSO %ADBL DB2LLIB(''DSNA.SDSNEXIT DSNA.SDSNLOAD'')`

    - If you chose option C, issue the following command to run the ADBRUN CLIST that you created:

      `%ADBRUN DB2LLIB(''DSNA.SDSNEXIT DSNA.SDSNLOAD'')`

    If more than one DB2 subsystem is active, the Active DB2 Systems panel (ADB2SYS) that shows all DB2 subsystems, as shown in the following figure, is displayed.

```
DB2 Admin ------------------ Active DB2 Systems -------------- Row 1 from 18
Command ===>                                            Scroll ===> PAGE


This is a list of the active DB2 systems on this MVS system.

Enter:
DB2 system name ===> DB2X          Retain DB2 system name ===> YES  (Yes/No)

Or select the one you wish to use, or press END to exit.

Sel DB2 System Description                                           Group
--- ---------- -------------------------------------------------- -----
    DB2A       Basic system 1
    DB2B       Local business system
    DB2C       Data sharing system
```
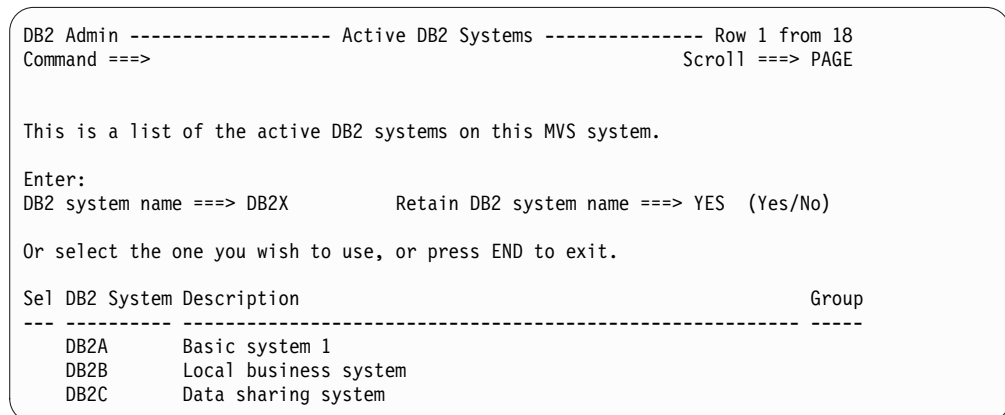
Figure 15. The Active DB2 Systems panel (ADB2SYS)

4. Select the DB2 subsystem that you want to use and press Enter. The DB2 Admin Main Menu panel, as shown in the following figure, is displayed.

```
DB2 Admin -------------- DB2 Administration Menu 11.2.0 ------------------ 00:49
Option ===> 1

   1 - DB2 system catalog                              DB2 System: DB2X
   2 - Execute SQL statements                          DB2 SQL ID: ISTJE
   3 - DB2 performance queries                         Userid    : ISTJE
   4 - Change current SQL ID                           DB2 Rel   : 1115
   5 - Utility generation using LISTDEFs and TEMPLATEs
   P - Change DB2 Admin parameters
  DD - Distributed DB2 systems
   E - Explain
   Z - DB2 system administration
  SM - Space management functions
   W - Manage work statement lists
   X - Exit DB2 Admin
  CC - DB2 catalog copy version maintenance
  CM - Change management


Interface to other DB2 products and offerings:
   I  DB2I   DB2 Interactive
   C  DB2 Object Comparison Tool
```

Figure 16. The DB2 Admin Main Menu panel (ADB2)

> **Attention:** If an SQL error occurs, ensure that the application plan (ADB) and the packages (ADBMAIN, ADB2GET, and ADB2CON) are bound correctly on the DB2 subsystem that you are using. Also, verify that you have granted EXECUTE on the application plan ADB to the user IDs that are running DB2 Admin. Restart from 2 on page 107.

5. Verify that DB2 Admin is available with limited functionality by completing the following steps:

   a. Select option **1** to go to the DB2 System Catalog.

   b. Select option **T** to go to the Tables, Views, and Aliases panel. Specify SYSDUMMY1 for the name and specify SYSIBM for the owner.

   c. Verify that DB2 Admin can retrieve rows from the DB2 catalog. You should see a row that describes the table SYSDUMMY1 owned by user SYSIBM.

   d. Use the DDL line command on the row for SYSDUMMY1. You should see a CREATE TABLE statement that can be used to create the table.

Example

To invoke DB2 Admin for DB2 subsystem ABCD, enter the following command on the command line of the ISPF main menu:

```
TSO %ADBL SYSTEM(ABCD)
```

To invoke the DB2 Admin Launchpad panel, which enables you to continue using DB2 Admin functions or to invoke another DB2 tool, enter the following command on the command line of the ISPF command processor panel (usually ISPF option 6):

```
%ADBL DMT
```

To directly invoke DB2 Admin and display all of the active DB2 subsystems that are available to you, enter the following command on the command line of the ISPF command processor panel (usually ISPF option 6):

```
%ADBL SHOW
```

Optional: Customize the ADB2UCUS

The data set names used in various jobs are set through the use of ISPF skeleton members. The data set name defaults are provided in skeleton ADB2UCUU, which is stored in the SADBSLIB library. Skeleton ADB2UCUS imbeds skeleton ADB2UCUU.

DB2 Admin provides you with the ability to align the product with your local data set and utility ID (UID) naming conventions.

To do so, modify the ADBU002 usermod, which resides in the SADBSAMP library. Copy desired lines from the ADB2UCUU skeleton to the ADBU002 usermod and modify as needed. All customization in skeleton ADB2UCUS should be done after it imbeds skeleton ADB2UCUU, unless otherwise noted.

When you subsequently run SMP/E to receive and apply SMP/E usermod ADBU002, the updated ISPF JCL skeletons will be added to the SADBSLIB library.

The following variables are some of the variables that are available for use (see skeleton member ADB2UCUT for a complete list):

Variable:
   Description:

&AJDATE
   Julian date (YYDDD)

&AJDAY
   Julian day (DDD)

&AYEAR4
   4-digit year (YYYY)

&AGDATE
   Gregorian date (YYMMDD)

&ANMON
   Numeric month (MM)

&ADAY
   Day (DD)

&AYEAR
   2-digit year (YY)

&ACMON
   3-character month (XXX)

&ATIME
   Time (HHMMSS)

&ATIME7
   Time with tenths of seconds (HHMMSST)

&ATIME4
   Time without seconds (HHMM)

&AHOUR
   Hour (HH)

&AMIN
   Minute (MM)

&ASEC
   Seconds (SS)

All lines that might require tailoring are preceded by SET statements (indicated by
)SET).

Example: This example demonstrates several different types of data set naming
changes using the variable ASYCPY1:

The variable ASYCPY1 is shipped as:
```
)SET ASYCPY1 = &PREFIX..&DB2SYS..IC.&DBNAME..&NAME.(+1)
```

To change the high-level qualifier from the current TSO PREFIX to MYHLQ, specify:
```
)SET ASYCPY1 = MYHLQ.&DB2SYS..IC.&DBNAME..&NAME.(+1) /* CHANGE HLQ TO FIXED STRING
```

To change the second-level qualifier from the DB2 subsystem ID to TEST, specify:
```
)SET ASYCPY1 = &PREFIX..TEST.IC.&DBNAME..&NAME.(+1)  /* CHANGE SUBSYSTEM TO 'TEST'
```

To insert a high-level qualifier of MYHLQ in front of the current TSO PREFIX and to
remove the DB2 database name, specify:
```
)SET ASYCPY1 = MYHLQ.&PREFIX..&DB2SYS..IC.&NAME.(+1)
               /* CHANGE HLQ TO FIXED STRING,
               /* INCLUDE PREFIX, REMOVE DBNAME
```

To use sequential data sets rather than a GDG data set, specify a data set name
that contains date and time values to generate unique data set names:
```
)SET ASYCPY1 = &PREFIX..IC.&DBNAME..&NAME..D&AJDATE..T&ATIME
```

Example: This example demonstrates several different types of utility ID (UID)
naming changes using the variables PREFXUID, LOADUID, and UNLODUID.

The variables PREFXUID, LOADUID, and UNLODUID are included as:
```
)SET PREFXUID = &Z
)SET LOADUID = &PREFXUID
)SET UNLODUID = &PREFXUID
```

To change the LOAD and UNLOAD UIDs such that they contain the TSO user ID,
a time stamp, and a utility type identifier, specify:
```
)SET PREFXUID = &ZUSER.&ATIME
)SET LOADUID = &PREFXUID.LD
)SET UNLODUID = &PREFXUID.UL
```

This setup sets the value of LOADUID to &ZUSER.&ATIME.LD and UNLODUID to
&ZUSER.&ATIME.UL. So, if the user ID is 'JOE' and the JCL for the LOAD utility is
generated at time '095344', the UID in the JCL for the LOAD utility is set to
'JOE095344LD'.

The maximum size of &ZUSER is 8 bytes, the size of &ATIME (HHMMSS) is 6
bytes, and the size of the literal is 2 bytes. The total maximum size is 16 bytes,
which is the maximum UID size.

To change the LOAD and UNLOAD UIDs such that they contain the TSO user ID
and a time stamp with tenths of seconds (USERID.HHMMSST), specify:
```
)SET PREFXUID = &ZUSER..&ATIME7
)SET LOADUID = &PREFXUID
)SET UNLODUID = &PREFXUID
```

This setup sets the value of LOADUID and UNLODUID to &ZUSER..&ATIME7. So, if the user ID is 'JOE' and the JCL for the LOAD utility is generated at time '0953446', the UID in the JCL for the LOAD utility is set to 'JOE.0953446'.

The maximum size of &ZUSER is 8 bytes, the size of a period is 1 byte, and the size of the &TIME7 (HHMMSST) is 7 bytes. The total maximum size is 16 bytes, which is the maximum UID size.

Restrictions:
- When modifying data set names, be sure that no data set names run beyond column 71 in the ADB2UCUS data set. Any characters beyond column 71 are truncated.
- Data set names, including the periods, cannot be greater than 44 bytes in length. Be sure that generated data set names are not longer than 44 bytes.
- Utility IDs (UIDs), including the periods, cannot be greater than 16 bytes in length. Be sure that generated UIDs are not longer than 16 bytes.
- Utility ID (UID) customization does not apply to UIDs in work statement lists (WSL).

For testing purposes, copy the ADB2UCUS skeleton to a private skeleton library and make your changes. This private skeleton library must first be allocated in the ISPSLIB concatenation (using the USERADD parameter of the ADBL CLIST).

After testing is complete, you can use an SMP/E USERMOD to update the DB2 Administration Tool V11.2 - product libraries. A sample SMP/E USERMOD is provided in member ADBU002 in the SADBSAMP library. Instructions for completing this step are provided in sample job ADBU002.

Optional: Tailor the DB2 Admin Launchpad

The DB2 Admin Launchpad enables you to launch all installed IBM DB2 tools that have an ISPF interface directly from a centralized panel.

Procedure
1. Run the ADBL CLIST with the DMT option, which creates the Launchpad table.
2. Perform the steps in the following topic: "Required in some cases: Update the APF Authorization table" on page 98

Optional: Grant SELECT access on catalog tables

DB2 Admin uses dynamic SQL against the catalog.

If you plan to make DB2 Admin available to a large number of users, you might want to specify those IDs that are authorized to see the catalog. To complete this step, run the Tools Customizer job with the template ADBGC.

Optional: Define Reverse Engineering stored procedure for CC/390

You can apply Reverse Engineering to additional software products such as Control Center OS/390 (CC/390).

Complete the following steps:

| 1. Run job ADBREST. ADBREST creates the Reverse Engineering stored procedure ADB2RE. ADB2REST also creates the required temporary tables and bind package for the stored procedure on the DB2 subsystem that will use the Reverse Engineering facility.
| 2. Copy load module members ADB2RE and ADB2LM from the SADBLLIB load library to one of the libraries defined as STEPLIB in the Work Load Manager (WLM)-managed stored procedure address space. The ADB2RE stored procedure dynamically allocates output data sets, as specified by the CC/390 user.

| The ADB2RE stored procedure must be defined with the SECURITY USER clause. Procedures that are specified with the SECURITY USER clause cannot run in the non-WLM-managed stored procedure address space (ssidSPAS). Instead, the stored procedure must run in a WLM-managed stored procedure address space.

Optional: Optimize DSNWZP and DSNZPARM settings

DB2 Admin Reverse Engineering uses the DSNWZP stored procedure to read values from DSNZPARM.

If DSNWZP does not complete normally, Reverse Engineering waits until DSNWZP times out. In this case, the GEN or DDL commands might run longer than necessary as a result of DB2 waiting for the timeout value for stored procedures to be reached.

To optimize performance, verify that the DSNWZP stored procedure is operational and that the DSNZPARM STORTIME(DSN6SYSP) parameter is set at a proper level. You might want to reduce the value specified for the DSNZPARM STORTIME(DSN6SYSP) parameter.

Optional: Enabling DB2 Admin distributed support

You can use DB2 Admin on remote DB2 systems. This functionality is called distributed support.

About this task

On remote systems, you can perform the following tasks through DB2 Admin:
- Build utility jobs and submit them to run on remote systems
- Perform alter and migrate functions for remote systems
- Issue SQL statements against remote systems
- Issue distributed GRANT and REVOKE commands
- Issue other commands on remote systems

By using distributed support and the Change Management functionality, you can register a multi-target change on a target system using DRDA access.

You partially enable distributed support when you customize DB2 Admin with Tools Customizer. To completely enable distributed support, complete the following procedure.

Procedure

Copy the appropriate load module to the load data set.

Option	Description
To enable distributed support:	Copy the distributed load module ADB2RCP to the load data set for the default stored procedure address space on the DB2 subsystem.
To enable distributed support and registration of a multi-target change registration on a target system using DRDA access:	Copy the load module ADBCRSP to the load data set for the default stored procedure address space on the DB2 subsystem.

Optional: Make DB2 Admin available to users

You can make DB2 Admin available to users.

Use one of the following methods to improve performance when invoking DB2 Admin:
- Copy the DB2 Admin ISPF and TSO libraries to your standard libraries. Your standard libraries are allocated in your TSO LOGON procedure or are allocated dynamically before you invoke ISPF.
- Allocate the DB2 Admin target libraries in the TSO LOGON procedure or dynamically before you invoke ISPF.

Using one of these methods eliminates the need for performing ISPF LIBDEFs each time that DB2 Admin is invoked, and significantly reduces DB2 Admin start-up time.

Tip: If possible, define the libraries that you are using for DB2 Admin (and all of the libraries allocated on the same DD statements before the ones that you are using for DB2 Admin) to LLA with the FREEZE option. This approach will significantly reduce the number of input/outputs (I/Os) and the I/O time used when ISPF and TSO perform a search for DB2 Admin members in the concatenation sequence.

To make DB2 Admin available while reducing the tailoring effort, use the ADBL CLIST to allocate the libraries, and invoke DB2 Admin by calling the ADB CLIST.

Optional: Making Object Comparison Tool available from DB2 Administration Tool

You can make the DB2 Object Comparison Tool available from DB2 Admin as part of the DB2 Admin customization process. You can also customize the Object Comparison Tool separately from the customization of DB2 Admin.

About this task

Follow the steps in the *IBM DB2 Administration Tool for z/OS User's Guide and Reference*, Chapter 2, *Starting and preparing Tools Customizer for use.*

Optional: Make the DB2I and Object Comparison Tool available from the DB2 Administration Tool

You can make the DB2I and Object Comparison Tool available from the main menu of the DB2 Admin Tool.

Before you begin

- All of the product customization steps that must be done before Tools Customizer is started are complete.
- The LPAR ISPF libraries that are required to submit the jobs are known.
- Tools Customizer is started.
- The Tools Customizer settings have been reviewed or modified, and saved.

About this task

DB2 Interactive (DB2I) is a DB2 facility that enables you to perform most DB2 tasks interactively.

DB2 Object Comparison Tool is an Administration Tool extension that lets you compare source and target objects, and generate reports that show the differences between the objects. The tools can also generate the jobs that are required to apply changes to the target.

When you customize DB2 Admin Tool for the first time or recustomize it, you can add DB2I and Object Comparison Tool as options in the DB2 Administration Menu as shown in the following figure under the section **Interface to other DB2 products and offerings**.

```
DB2 Admin -------------- DB2 Administration Menu 11.2.0 ------------------ 00:49
Option ===> 1

   1 - DB2 system catalog                              DB2 System: DB2X
   2 - Execute SQL statements                          DB2 SQL ID: ISTJE
   3 - DB2 performance queries                         Userid    : ISTJE
   4 - Change current SQL ID                           DB2 Schema: ISTJE
   5 - Utility generation using LISTDEFs and TEMPLATEs DB2 Rel   : 1115
   P - Change DB2 Admin parameters
  DD - Distributed DB2 systems
   E - Explain
   Z - DB2 system administration
  SM - Space management functions
   W - Manage work statement lists
   X - Exit DB2 Admin
  CC - DB2 catalog copy version maintenance
  CM - Change management

Interface to other DB2 products and offerings:
   I  DB2I   DB2 Interactive
   C  DB2 Object Comparison Tool
```

Figure 17. DB2 Administration Menu (ADB2)

If you ran the DB2 Admin Discover EXEC, you must review the values that were discovered.

Procedure

1. Specify E next to the **Product parameters** field on the Customizer Workplace panel, and press Enter. The Product Parameters panel is displayed as shown in the following figure.

```
CCQPPRD                      Product Parameters                      17:17:35
 Command ===>                                               Scroll ===> CSR

 Complete the following tasks to customize the products. The required tasks,
 required steps within a required or selected task, and required parameters
 are preceded by an asterisk (*). Ensure that all values are specified for
 the required parameters. Press End to save and exit.

 Commands: SAVE - Save parameter values
 Line Commands: / - Select

 Product customization library .: ADB.TCZ.BETA.CUST.$SY4A$.ADB1020
                                                             More:   - +
    Option 1: Menu option . . . . . . . . . . . . . . . I                   >
          Description . . . . . . . . . DB2I                                >
    ISPF statement . . . . SELECT CMD(%DSNECPRI SSID(&DB2SYS))NEWAPPL(DSNE)PASSLIB   >
    ISPF panel . . . . . . . . . . . . . .
    SQL statement . . . . . . . . . . . . .                                 >
    Admin Tool command . . . . . . . . . .                                  >
    New DB2 attachment for option 1 . . . .      (YES, NO)
    Option 2: Menu option . . . . . . . . . .     C                         >
          Description . . . . . . . . . .      DB2 Object Comparison Tool   >
    ISPF statement . . . . . . . . . . . . .                                >
    ISPF panel . . . . . . . . . . . . . .        GOCMENU
    SQL statement . . . . . . . . . . . . .                                 >
    Admin Tool command . . . . . . . . . .                                  >
    New DB2 attachment . . . . . . . . . .        (YES, NO)
    Option 3: Menu option . . . . . . . . .                                 >
          Description . . . . . . . . . .                                   >
    ISPF statement . . . . . . . . . . . . .                                >
    ISPF panel . . . . . . . . . . . . . .
    SQL statement . . . . . . . . . . . . .                                 >
    Admin Tool command . . . . . . . . . .                                  >
    New DB2 attachment . . . . . . . . . .        (YES, NO)
```

Figure 18. Product Parameters panel

2. Set the DB2 Admin main menu options for DB2I.

   You might have to scroll through several pages before you find these options.

   a. Specify a value for the option, **Option 1**.

      Tip: This value is displayed in DB2 Administration Menu so you want your
      users to associate this value with invoking DB2I. I is a logical choice.

   b. Specify a value for the option, **Option 1 description**.

      Tip: This value describes option 1, so DB2I is a good choice.

   c. Specify SELECT CMD(%DSNECPRI SSID(&DB2SYS)) NEWAPPL(DSNE) PASSLIB for
      the expanded value for the option, **ISPF statement for Option 1**.

3. Set the DB2 Admin main menu options for the Object Comparison Tool.

   a. Specify a value for the option, **DB2 Admin command for Option 2**.

      Tip: This value is displayed in DB2 Administration Menu so you want your
      users to associate this value with invoking the Object Comparison Tool. C is
      a logical choice.

   b. Specify a value for the option, **Option 2 description**.

      Tip: This value describes option 2, so DB2 Object Comparison Tool is a
      good choice.

   c. Specify GOCMENU for the option, **ISPF panel for option 2**.

4. Generate the customization jobs for the DB2 subsystems (SSIDs) on which you
   want to have DB2 Interactive and DB2 Object Comparison Tool.

5. Submit the ADBCUST job for each of the DB2 subsystems that you applied a customization job to.

Chapter 4. Using the DB2 Admin Launchpad

The DB2 Admin Launchpad provides a convenient way to run DB2 tools.

The topics in this information describe how to prepare and use the DB2 Admin Launchpad. Use the DB2 Admin Launchpad to launch installed IBM DB2 tools directly from a centralized panel. When you launch a tool, you are presented with the tool's first panel.

Restriction: Only tools that have an ISPF interface can be launched from the DB2 Admin Launchpad function.

Using the Launchpad consists of the following steps:
- Create an ISPF table that contains an entry for each tool you want to launch.
- Modify the ISPF table to add, delete, or update tool entries.
- Launch the tools by displaying the ISPF table and selecting the tools.

Topics:
- "Step 1. Create the Launchpad table"
- "Step 2. Modify the Launchpad table" on page 120
- "Step 3. Launch tools" on page 124

Step 1. Create the Launchpad table

The table is created the first time you run the ADBL CLIST with the DMT option, and resides in the table library data set.

The Launchpad table, named ADBDMT, must contain an entry for each tool that you want to launch. After the table is created, you need to populate it with the tools that you want to launch from the launchpad.

After you have created the Launchpad table, run the ADBL CLIST with the DMT parameter to display it. You invoke the launchpad from standalone TSO, or from ISPF panel 6, the TSO Command Panel. You enter a command similar to the following example:

```
EX 'ADBA2MPE.SADBCLST(ADBL)' 'PRODADD(GOCB10)
LIBAPRE(SGOC) PROD(ADBB10) LIBPRE(SADB) DMT'
```

The following figure shows the table immediately after it has been created.

```
DB2 Admin ------------------- DB2 Tools Launchpad ------------------ Row 1 from 8
Command ===>                                            Scroll ===> PAGE

Specify DB2 SSID (opt) ===>        (Enter '?' for a list of active SSIDs)

Select the DB2 tool you wish to launch or enter its code in the command line.

Sel Code  Tool Name                                      Rel  Prog No.
    ---   ------------ ADMINISTRATION TOOLS -------------  ---  --------
    ADM   DB2 Administration Tool                          810  5697-L90
    ---   --------- APPLICATION MANAGEMENT TOOLS --------  ---  --------
          No table entries in this category
    ---   --------- PERFORMANCE MANAGEMENT TOOLS --------  ---  --------
          No table entries in this category
    ---   -- RECOVERY AND REPLICATION MANAGEMENT TOOLS --  ---  --------
          No table entries in this category
****************************** Bottom of data ********************************
```

Figure 19. Launchpad Table panel (ADBDMT)

This panel groups the DB2 tools into the following four categories:
- Administration
- Application Management
- Performance Management
- Recovery and Replication Management

These categories make it easier to locate a tool on the panel. The following fields are shown on this panel:

Specify DB2 SSID (opt)
> You can specify a valid DB2 SSID, which makes it available to any and all tools that are invoked from the Launchpad. The SSID is stored in variable DMTSSID. The last SSID specified persists across ISPF sessions.

Sel   This column is used to specify the following actions that you wish to perform:
- ADD to add a new entry
- DEL to delete the entry on that row
- UPD to update the entry on that row
- S or / to start the tool

Code   Enter the tool code on the command line at the top of the panel.

Tool name
> The name of the tool.

Rel   The release or version number of the tool.

Prog No.
> The IBM program number of the tool.

Step 2. Modify the Launchpad table

You can use one of two methods to add, delete, or update entries in the Launchpad table.

You can modify the Launchpad table by using one of the following two methods:
-

  Dialog method
  > The dialog method consists of displaying the Launchpad table by using

the ADBL CLIST with the DMT parameter, and then entering the ADD, DELETE, or UPDATE command in the Sel column.

-

ADBDMTI EXEC method

Invoke the ADBDMTI EXEC with the ACTION parameter, with its values ADD, UPDATE and DELETE. Additional values correspond to the fields on the Launchpad Entry panel; this panel is displayed when you run the ADBDMTI EXEC. In general, it is easier to enter these values directly on the panel.

PID

This is the **program number** of the tool.

REL

This is the **release number** of the tool. When using several releases of the same tool, use utmost caution, so as not to get confused. Also, it is recommended that you assign them unique codes.

NAME

The name of the tool.

CDE

An arbitrary **code** used to identify or invoke the tool.

GRP

The **group** number used for grouping the tools on the panel.

STAT

This field indicates the **installed** status of the tool, and can have a value of Y or N.

CMD

Use this field to enter an ISPF string used to launch the DB2 tool. It is probably easier to specify the ISPF string directly in the **Command** field of the Launchpad Entry panel.

These values are discussed further on in this chapter - or on the Help panels associated with the Launchpad Entry panel; at this point, a brief example showing that the tool with product id 5655-D38 is to be deleted, should give you an idea of the ADBDMTI interface to the Launchpad.

```
ADBDMTI ACTION(DELETE) PID(5655-D38)
```

To update or delete a table entry, you must provide a PID number (with or without a Rel identifier), a code or a name.

Adding tools to the Launchpad table

You can add a tool to the Launchpad table by using both the dialog method and the ADBDMTI EXEC method.

Using the dialog method to add tools to the Launchpad table Procedure

1. Use the ADBL CLIST with the DMT parameter. The Launchpad Table panel is displayed.
2. Specify ADD in the Sel column of any row. The Launchpad Entry panel is displayed, as shown in the following figure.

```
DB2 Admin ---------- DB2 Tools Table - ADD An Entry ------------------------------
  Command ===>


  Library : USER01.ISPF.ISPTLIB

  Tool Name : DB2 Object Comparison Tool for z/OS
  Code      : OBJ        (User-defined code, for shortcut tool identifier)
  Prog No.  : 5655-DOC   (IBM program product number or equivalent)
  Release   : 111        (Product release number)
  Group     : 1          (Tool category, as follows:
                             1 - Administration Tools
                             2 - Application Management Tools
                             3 - Performance Management Tools
                             4 - Recovery and Replication Management)
  Installed : Y          (Yes/No)

  Command   : SELECT MODE(FSCR) CMD(%ADB PANEL(GOCMENU))
```

Figure 20. Launchpad Entry panel (ADBDMTI)

3. Specify any additional information that identifies the tool. You can specify information in the following fields:

Tool Name
Enter the name of the DB2 tool with which you want to work.

Code
Enter a user-defined shortcut name to identify a tool. Code values should be unique. Although it is possible to duplicate code values, doing so could result in DB2 running the wrong tool when the code is invoked.

Prog No.
Enter the IBM program product number or equivalent.

Release
Enter the release/version number of the tool.

Group
Specify the group in which the tool belongs. These groups help to make all the tools easier to locate on the display panel. The following values are permissible:
- 1 - Administration Tools
- 2 - Application Management Tools
- 3 - Performance Management Tools
- 4 - Recovery and Replication Management Tools

Installed
Indicate whether the tool is installed or not. If the status of the tool is N (not installed), you can create a table entry for it; however, this table entry is not displayed on the panel. If you install the tool later and want to include it on the Launchpad display panel, use the ADBDMTI EXEC to change the N to Y.

Command
Enter an ISPF string used to launch the DB2 tool. This field does not require continuation characters for very long command strings, as it accepts free-form format that wraps to the next line.

Help panels provide additional information about these input fields. The tool that you specified is added.

Using the ADBDMTI EXEC to add tools to the Launchpad table
Procedure

1. Invoke the ADBDMTI EXEC that includes the ACTION(A) or ACTION(ADD) parameter. Because ADD is the default, you can omit this parameter. The following examples show how to use the ADBDMTI EXEC to add tools:
   - `ADBDMTI ACTION(A)`
   - `adbdmti action(add) CDE(OBJ) pid(1234-567) name(OBJECT COMPARISON) rel(565) stat(Y) grp(1) cmd(ex 'dsn.support.clist')`

   The Launchpad Entry panel, as shown in the previous figure, is displayed. Any values that you specified on the ADBDMTI statement are used to fill in the panel.

2. Specify any additional information that identifies the tool. For types of information that you can specify, see Types of information that identify DB2 tools. The tool that you specified is added.

Updating tools in the Launchpad table

You can use the dialog method or the ADBDMTI EXEC method to update tools in the Launchpad table.

Using the dialog method to update tools in the Launchpad table
Procedure

1. Use the ADBL CLIST with the DMT parameter. The Launchpad Table panel is displayed.
2. Specify UPD in the Sel column of any row. The Launchpad Entry-Update panel is displayed.
3. Overwrite the information that you want to modify and press Enter. The entry in the Launchpad table is updated.

Using the ADBDMTI EXEC method to update tools in the Launchpad table
Procedure

1. Invoke the ADBDMTI EXEC that includes the ACTION(U), ACTION(UPD), or ACTION(UPDATE) parameter and identify the tool by specifying its name, code, or PID number. The following example shows how to use the ADBDMTI EXEC to update tools:

   `ADBDMTI ACTION(UPDATE) CDE(OBC)`

   The Launchpad Entry-Update panel is displayed.
2. Overwrite the information that you want to modify and press Enter. The entry in the Launchpad table is updated.

Deleting tools from the Launchpad table

You can use the dialog method or the ADBDMTI EXEC method to delete tools from the Launchpad table.

Using the dialog method to delete tools from the Launchpad table
Procedure

1. Use the ADBL CLIST with the DMT parameter. The Launchpad Table panel is displayed.
2. Specify DEL in the Sel column of the appropriate row. The Launchpad Entry-Delete panel is displayed.

3. Confirm whether to delete the specified tool from the table.
   - Specify Y to delete the tool.
   - Specify N or press End to cancel the delete operation.

Using the ADBDMTI EXEC method to delete tools from the Launchpad table
Procedure
1. Invoke the ADBDMTI EXEC that includes the ACTION(D), ACTION(DEL), or ACTION(DELETE) parameter. The following example shows how to invoke the ADBDMTI EXEC to delete tools:

   ```
   ADBDMTI ACTION(DELETE) CDE(OBC)
   ```
2. Confirm whether to delete the specified tool from the table.
   - Specify Y to delete the tool.
   - Specify N or press End to cancel the delete operation.

Step 3. Launch tools

You can launch DB2 tools by using one of two methods.

About this task

To launch DB2 tools:

Procedure
1. Use the ADBL CLIST with the DMT parameter. The Launchpad Table panel is displayed.
2. Use either of the following methods to launch a tool:
   - Enter an S or a slash (/) in the Sel column.
   - Enter the code associated with the tool on the command line and press Enter.

     Important: When you enter a code, make sure that the code is unique because the results are unpredictable if multiple tools have the same code.

Chapter 5. Using DB2 Admin panels

The topics in this information explain how to use DB2 Admin panels.

The release level and mode of your DB2 subsystem affect the options that are available to you from the panels.

Topics:

Types of DB2 Admin panels

DB2 Admin uses three types of panels.
- Table display panels
- BROWSE panels
- SQL error display panels

Using table display panels

Table display panels contain ISPF tables that show information about DB2 objects.

You use table display panels to access DB2 Admin functions.

Note: DB2 Admin panels might hide fields or make entry fields output-only in some cases (for example, if a version of a DB2 is used that has this restriction). Hidden fields cause the appearance of blank lines or spaces, but you can disregard these blank lines or spaces. Note that:

- Data entry fields, both the description preceding the entry field and the input field, can be entirely hidden,
- Data entry fields can be output-only, which means that you can see that there is a field there (because the description is visible) but you cannot provide a value.
- A column on a table display can be converted to output-only, or even hidden.

The panel in the following figure, is an example that shows the areas on a typical table display panel.

```
          DB2 Admin ------------------ DB2X Databases --------------- ROW 1 TO 5 OF 5
A ──►    Command ===>                                          Scroll ===> PAGE

       ┌ Valid line commands are:
       │  T - Tables  S - Table spaces  X - Indexes  G - Storage group  ICS - IC status
B ──►  │  DIS - Display database  STA - Start database  STO - Stop database  A - Auth
       └  ? - Show all line commands
                                    Storage Buffer           Created  Read Only Share  ┐
          Select Name     Owner     Group   Pool      DBID By         S Timestamp       ├─ C
                 *        *         *       *         *      *         * *  ◄──────────── D
          ------ -------- -------- -------- -------- ------ -------- - ------------------
E ──►            D        ISTJE2   G        BP0        293 ISTJE2     0001-01-01-00.00.0  ┐
          *IS    ISTJE2DC ISTJE2   ISTJE2GC BP0        295 ISTJE2     0001-01-01-00.00.0  │
          =      ISTJE2DE ISTJE2   ISTJE2GE BP0        269 ISTJE2     0001-01-01-00.00.0  ├─ F
                 ISTJE2DS ISTJE2   ISTJE2G  BP0        296 ISTJE2     0001-01-01-00.00.0  │
                 ISTJE2DV ISTJE2   ISTJE2G  BP0        294 ISTJE2     0001-01-01-00.00.0  ┘
G ──►    ****************************** END OF DB2 DATA *******************************
```

Figure 21. Table Display panel layout

First row of the panel

> Contains the DB2 Admin panel name and a count of data rows. The row count reflects an initial search done by your DB2 subsystem.

A

> **Command line.**
>
> > On this line, you can enter any DB2 command, ISPF command, or DB2 Admin primary command.

B

> **Line command description area.**
>
> > This area indicates the DB2 Admin line commands that you can issue from a particular table display panel. You issue a line command in the Select field (area E). When there is not enough space on a panel to list all valid line commands, only the most frequently used line commands are shown. To display all other valid line commands, specify a question mark (?) in the Select field, and press Enter.

C

> **Column headers.**
>
> > This area contains the names of the columns that contain data.

D

> **Search arguments.**
>
> > Use this area to enter search criteria for the data that is displayed in the panel. IPSF generic search argument rules apply in this area. For columns that contain alphabetic characters, the asterisk (*) under the column name marks the beginning (left-justified) of the area in which you can enter search criteria to limit the information that DB2 Admin returns. For columns that contain numeric characters, the asterisk (*) marks the end (right-justified) of the area. For example, you can enter D050 in the Name column to display only those databases whose names begin with D050.

E

Select column.
Use the Select column to issue DB2 Admin line commands (shown in area B) against DB2 objects that are listed in the Table Display panel.

F

Rows returned.
This area shows the rows that DB2 returns to you based on the options that you selected, the commands that you issued, or the search criteria that you entered. For example, to display the panel shown in the previous figure, request (on the System Catalog Menu panel) that all databases owned by ISTJE2 be displayed.

G

End of data marker.
This line indicates the end of the data returned from DB2.

If you enter a line command or update a row in the table display and also issue a scroll request (PF7 to scroll up or PF8 to scroll down), the line command or row update is processed and the scroll request is ignored.

Using BROWSE panels

BROWSE panels contain details about DB2 objects.

Issue the DB2 Admin BROWSE primary command from any table display panel to display the associated BROWSE panel for the object.

The panel in the following figure shows the BROWSE command being entered in a table display panel of tables stored in the DB2 catalog.

```
DB2 Admin ----------- DB2X Tables, Views, and Aliases ---------- Row 32 of 160
Command ===> BROWSE                                         Scroll ===> PAGE

Commands: GRANT  MIG
Line commands:
 C - Columns  A - Auth  L - List  X - Indexes  S - Table space  D - Database
 V - Views  T - Tables  P - Plans  Y - Synonyms  SEL - Select prototyping
 ? - Show all line commands

Sel   Name               Owner    T DB Name  TS Name    Cols      Rows Checks
      *                  *        * *        *           *          *      *
----- ------------------ -------- - -------- -------- ------ ----------- ------
      EACT               DSN8810  T DSN8D81A DSN8S81R      5          -1      0
      EPROJACT           DSN8810  T DSN8D81A DSN8S81R      7          -1      0
      EEPA               DSN8810  T DSN8D81A DSN8S81R      8          -1      0
      VPHONE             DSN8810  V DSN8D81A DSN8S81E      7          -1      0
      VEMPLP             DSN8810  V DSN8D81A DSN8S81E      2          -1      0
```

Figure 22. Issuing the DB2 Admin BROWSE primary command on the Tables, Views, and Aliases panel (ADB21T)

The BROWSE primary command generates output similar to that shown in the following figure. Output is in ISPF browse format. The first line is a header with the DB2 column names. To display the remaining columns, scroll to the right.

```
DB2 Admin ----- DB2X Browse Result of SQL Select ---- Line 00000000 Col 001 080
 Command ===>                                                  Scroll ===> PAGE

******************************* Top of Data **********************************
 NAME               CREATOR  TYPE DBNAME   TSNAME    DBID   OBID COLCOUNT EDPRO
 ------------------ -------- ---- -------- -------- ------ ------ -------- -----
 DSNRLST01          SYSIBM   T    DSNRLST  DSNRLS01   256     3      11
 DSN_REGISTER_APPL  DSNRGCOL T    DSNRGFDB DSNRGFTS   257     3       9
 DSN_REGISTER_OBJT  DSNRGCOL T    DSNRGFDB DSNRGFTS   257     6      11
 DEPT               DSN8810  T    DSN8D81A DSN8S81D   258    11       5
 VDEPT              DSN8810  V    DSN8D81A DSN8S81D     0     0       4
```

Figure 23. Output from the BROWSE command

DB2 Admin can also display data in tables that contain binary large objects
(BLOBs), character large objects (CLOBs), double-byte character large objects
(DBCLOBs), and ROWID columns:
- For BLOBs, DB2 Admin retrieves up to 128 bytes per column and displays the
  data in hexadecimal format.
- For CLOBs, DB2 Admin retrieves up to 256 bytes per column and displays the
  data in character format.
- For DBCLOBs, DB2 Admin retrieves up to 128 bytes per column and displays
  the data in hexadecimal format.
- ROWIDs are displayed in hexadecimal format.

Using SQL error display panels

If an error occurs during running of an SQL statement, DB2 Admin displays the
SQL code and error message on a separate panel called an SQL error panel.

To correct the SQL statement, press END, which redisplays the panel where you
originally issued the SQL statement. DB2 Admin positions the cursor at the point
in the SQL statement where DB2 found the error.

The following figure shows the error panel that DB2 Admin displays when the
following SQL statement (containing a spelling error) is issued: SELECT * FRON
Q.STAFF.

```
DB2 Admin -------------------- DB2 Error Display 1 -------------------- 14:14
 Command ===>
 Rollback done
    SQLCODE : -104                        DSNTIAR CODE :  0

 DSNT408I SQLCODE = -104, ERROR:  ILLEGAL SYMBOL FRON VALID SYMBOLS ARE FROM
          INTO
 DSNT418I SQLSTATE   = 37501 SQLSTATE RETURN CODE
 DSNT415I SQLERRP    = DSNHPARS SQL PROCEDURE DETECTING ERROR
 DSNT416I SQLERRD    = 0  0  0  -1  10  0 SQL DIAGNOSTIC INFORMATION
 DSNT416I SQLERRD    = X'00000000'  X'00000000'  X'00000000'  X'FFFFFFFF'
          X'0000000A'  X'00000000' SQL DIAGNOSTIC INFORMATION
```

Figure 24. Error Display panel (part 1 of 2)

Press Enter to see error panel two, as shown in the following figure.

```
DB2 Admin -------------------- DB2 Error Display 2 -------------------- 14:14
 Command ===>

    SQLCODE : -104                          DSNTIAR CODE :  0

PREPARE

SELECT * FRON Q.STAFF
```

Figure 25. Error Display panel (part 2 of 2)

Press END to redisplay the panel in which you entered the incorrect SQL
statement.

Finding the source code for panels

Whenever DB2 Admin panels are discussed in this information, the name of the
panel in the figure caption is followed by another name in parentheses. The name
in parentheses is the source code panel name.

For example, in Figure 40 on page 141, the figure caption is "DB2 Administration
Menu Panel (ADB2)." ADB2 is the source code panel name.

If you are developing DB2 Admin applications, you can use the source code name
to quickly locate the source code for a specific panel.

To display the name of the panel in the upper left corner of the panel, issue the
ISPF command PANELID ON.

Using DB2 Admin commands

You can use two types of DB2 Admin commands.

You can use the following types of DB2 Admin commands:
• Primary commands
• Line commands

Primary commands

Primary commands can be issued from the command line on DB2 Admin panels.

Most primary commands can be entered on all panels; however, some primary
commands are restricted to certain panels.

For information on the syntax for primary commands, see the Help panels.
Related reference:
"DB2 Admin primary commands" on page 1037
Primary commands are issued from the command line on DB2 Admin panels.

Line commands

Line commands specify an operation that is to be performed on the information
that is displayed.

Line commands are issued from ISPF table display panels. Specify line commands
in the line command area in front of each row (called the SELECT field).

Two types of line commands are available:

- Special line commands
- General line commands

If you enter a line command or update a row in the table display and also issue a scroll request (PF7 to scroll up or PF8 to scroll down), the line command or row update is processed and the scroll request is ignored.

Special line commands

The special line commands that are available for a panel are listed in the line command description area.

A question mark (?) line command indicates that there is not enough room to show all line commands. Specify ?, to display a list of all valid line commands for that panel.

General line commands

Three general line commands are available: minus (-), equal (=), and slash (/).

Minus (-) line command

Use the - line command to exclude a line from a list on table display panels.

You can enter more than one - line command at a time.

Equal (=) line command

Use the = line command to repeat the last line command that you issued.

The panel in the following figure shows how the = line command is used. In this example, the DIS command is entered to request a display of the database named DBEDB1. When DB2 Admin returns from executing the line command, the asterisk replaces the first character of that command in the Select field. If you specify = in the Select field of the next line and press Enter, the DIS line command is executed for database DBEDB2.

Enter the = line command multiple times, as shown in Figure 27 on page 131, to issue the next line command when DB2 Admin returns from executing the current line command; the panel where the = line commands are entered is not shown between executions of the line commands.

```
DB2 Admin ------------------- DB2X Databases -------------------- Row 1 of 5
Command ===>                                                Scroll ===> PAGE

Commands: GRANT  MIG  DIS  STA  STO  UTIL
Line commands:
 T - Tables  S - Table spaces  X - Indexes  G - Storage group  ICS - IC status
 DIS - Display database  STA - Start database  STO - Stop database  A - Auth
 ? - Show all line commands
                          Storage Buffer          Created       Index
Select Name     Owner    Group    Pool     DBID By         T E BPool    I
       *        *        *        *         * *         * * *        *
------ -------- -------- -------- -------- ------ -------- - - -------- -
       ADBDCH   ADB      ADBGCH   BP1       271 ISTFL2       E BP2      Y
*IS    DBEDB1   DPGROTH  SYSDEFLT BP1       272 DPGROTH      E BP2      Y
=      DBEDB2   DPGROTH  SYSDEFLT BP1       273 DPGROTH      E BP2      N
       DSNDB04  SYSIBM   SYSDEFLT BP1         4 SYSIBM         BP2      N
       DSNDB06  SYSIBM                        6 SYSIBM       E BP0      N
****************************** END OF DB2 DATA ******************************
```

Figure 26. Issuing the '=' line command on the Databases panel (ADB21D)

```
DB2 Admin ------------------- DB2X Databases -------------------- Row 1 of 7
Command ===>                                                Scroll ===> PAGE

Commands: GRANT  MIG  DIS  STA  STO  UTIL
Line commands:
 T - Tables  S - Table spaces  X - Indexes  G - Storage group  ICS - IC status
 DIS - Display database  STA - Start database  STO - Stop database  A - Auth
 ? - Show all line commands
                          Storage Buffer          Created       Index
Select Name     Owner    Group    Pool     DBID By         T E BPool    I
       *        *        *        *         * *         * * *        *
------ -------- -------- -------- -------- ------ -------- - - -------- -
       ADBDCH   ADB      ADBGCH   BP1       271 ISTFL2       E BP2      Y
*IS    DBEDB1   DPGROTH  SYSDEFLT BP1       272 DPGROTH      E BP2      Y
*IS    DBEDB2   DPGROTH  SYSDEFLT BP1       273 DPGROTH      E BP2      N
       DSNDB04  SYSIBM   SYSDEFLT BP1         4 SYSIBM         BP2      N
=      DSNDB06  SYSIBM                        6 SYSIBM       E BP0      N
=      DSNDB07  DSCGDB2  SYSDEFLT BP1         7 ISTJE     W    BP2      N
       DSNRGFDB DSCGDB2  SYSDEFLT BP1       257 ISTJE        E BP2      N
****************************** END OF DB2 DATA ******************************
```

Figure 27. Issuing the '=' line command multiple times on the Databases panel (ADB21D)

Slash (/) line command

Use the / line command to show all column names and their values for the
selected row.

You can enter more than one / line command at a time.

The panel in the following figure illustrates the use of the / line command on
database DSNDB06.

```
DB2 Admin ------------------ DB2X Databases -------------------- Row 1 of 7
Command ===>                                           Scroll ===> PAGE

Commands: GRANT  MIG  DIS  STA  STO  UTIL
Line commands:
 T - Tables  S - Table spaces  X - Indexes  G - Storage group  ICS - IC status
 DIS - Display database  STA - Start database  STO - Stop database  A - Auth
 ? - Show all line commands
                          Storage  Buffer          Created        Index
Select Name     Owner    Group    Pool     DBID  By      T E BPool    I
       *        *        *        *         * *    * *    * * *      *
------ -------- -------- -------- -------- ------ -------- - - -------- -
       ADBDCH   ADB      ADBGCH   BP1         271 ISTFL2     E BP2      Y
*IS    DBEDB1   DPGROTH  SYSDEFLT BP1         272 DPGROTH    E BP2      Y
*IS    DBEDB2   DPGROTH  SYSDEFLT BP1         273 DPGROTH    E BP2      N
       DSNDB04  SYSIBM   SYSDEFLT BP1           4 SYSIBM       BP2      N
/      DSNDB06  SYSIBM                          6 SYSIBM     E BP0      N
       DSNDB07  DSCGDB2  SYSDEFLT BP1           7 ISTJE    W   BP2      N
       DSNRGFDB DSCGDB2  SYSDEFLT BP1         257 ISTJE      E BP2      N
****************************** END OF DB2 DATA ******************************
```

Figure 28. Issuing the '/' line command on the Databases panel (ADB21D)

The panel in the following figure shows the result. All column names and their
values from the catalog table (SYSIBM.SYSDATABASE in this case) are displayed.

```
DB2 Admin ------------------ DB2X Display Row ------------------- Row 1 of 18
Command ===>                                           Scroll ===> PAGE


S Column Name        Column Value
  *                  *
- ----------------- ---------------------------------------------------------
  NAME               DSNDB06
  CREATOR            SYSIBM
  STGROUP
  BPOOL
  DBID                      6
  IBMREQD            Y
  CREATEDBY          SYSIBM
  ROSHARE
  TIMESTAMP          0001-01-01-00.00.00.000000
  TYPE
  GROUP_MEMBER
  CREATEDTS          1985-04-01-00.00.00.000000
  ALTEREDTS          1985-04-01-00.00.00.000000
  ENCODING_SCHEME    E
  SBCS_CCSID                0
  DBCS_CCSID                0
  MIXED_CCSID               0
  INDEXBP            BP0
  IMPLICIT           Y
  CREATORTYPE
  RELCREATED         P
****************************** END OF DB2 DATA ******************************
```

Figure 29. Result of issuing the '/' line command

On the Launchpad panel (ADBDMT), you can issue / or s to invoke the ISPF
interface for the tool on that row. On the Launchpad panel, you can specify only
one / line command at a time.

Using the DB2 Admin Look Up function

Use the Look Up function to determine the valid values that you can enter in
certain input fields.

To use Look Up, type a question mark (?) in any field that is supported by Look Up and press Enter. (Not all input fields support Look Up; the fields that do support Look Up are denoted by a question mark at the end of the field.)

After you press Enter, a list of valid choices is displayed. You can select a value from this list by entering a plus sign (+) to the left of your choice.

You can also use Look Up with a qualifier. Enter the first few characters of a name followed by a question mark. To include all results containing the qualifier you are searching for, include the wildcard (%) with the qualifier. When you press Enter, all names that follow that naming convention are displayed. For example, TS01? finds all names that start with TS01, and %TS01? finds all names that contain TS01 in the name.

By using Look Up, you can save keystrokes and avoid typing errors. You can also avoid backing out of the current panel in order to search for the correct object.

Examples of using the DB2 Admin Look Up function

An example of using the DB2 Admin Look Up function is shown in the following figures. In this example, the DB2 Admin Look Up function is supported by two fields, TABLESPACE and IN, both of which show a question mark in the text to the right of the field. The table space name TSPACE01 has been entered, but the character string DSN? is a request to display all databases that begin with DSN.

```
ADB26CS n ------------------DB2X Create Table Space ------------------ 06:28
Command ===> _____

 CREATE

 TABLESPACE . . TSPACE01  (required table space name.  ? to look up)

 IN . . . . . . DSN?    (optional database. default=DSNDB04. ? to look up)

 Like:
 Database . . . _____ (optional existing database. ? to look up)
 Name . . . . . _____ (optional existing table space. ? to look up)
```

Figure 30. Using the DB2 Admin Look Up function — requesting a Look Up on the Create Table Space panel (ADB26CS)

The following figure shows the results of using Look Up. All databases that begin with DSN are displayed. Select an item by entering a plus sign (+) in the Select field next to the desired table entry. In this example, DSN8D81A is selected. When you press End, DB2 Admin enters this name in the IN field of the previous panel.

```
DB2 Admin ------------------ DB2X Databases --------------- Row 1 to 13 of 13
Command ===>                                              Scroll ===> CSR
Select by typing '+'
Commands: GRANT  MIG  DIS  STA  STO  UTIL
Line commands:
 T - Tables  S - Table spaces  X - Indexes  G - Storage group  ICS - IC status
 DIS - Display database  STA - Start database  STO - Stop database  A - Auth
 ? - Show all line commands
                         Storage  Buffer         Created        Index
Select Name     Owner    Group    Pool     DBID By        T E BPool    I
       *        *        *        *        * *         * * *          *
------ -------- -------- -------- -------- ------ -------- - - -------- -
       DSNDB04  SYSIBM   SYSDEFLT BP0          4 SYSIBM        BP0      Y
       DSNDB06  SYSIBM                         6 SYSIBM      E BP0      N
       DSNDB07  DB2ADM   SYSDEFLT BP0          7 DB2ADM    W   BP1      Y
       DSNDPSM  DB2ADM   SYSDEFLT BP0        293 DB2ADM      E BP1      N
       DSNRGFDB DB2ADM   SYSDEFLT BP0        257 DB2ADM      E BP1      N
       DSNRLST  DB2ADM   SYSDEFLT BP0        256 DB2ADM      E BP1      Y
       DSN7CDDB JAYANTI  CFCSG001 BP8K1      267 JAYANTI     E BP1      Y
       DSN7UDF  USRND05  DSN8G810 BP0        292 USRND05    U  BP1      Y
+      DSN8D81A DB2ADM   DSN8G810 BP0        259 DB2ADM      E BP1      Y
       DSN8D81E DB2ADM   DSN8G810 BP0        269 DB2ADM     U  BP1      Y
       DSN8D81L USRND05  DSN8G810 BP0        296 USRND05     E BP1      Y
       DSN8D81P DB2ADM   DSN8G810 BP0        268 DB2ADM      E BP1      N
       DSN8D81U DB2ADM   DSN8G81U BP0        270 DB2ADM      E BP1      N
```

Figure 31. Using the DB2 Admin Look Up function — selecting an object on the Databases panel (ADB21D)

For input fields that support more than one value, you can select multiple objects from the list by entering a plus sign next to each object that you want to select and pressing End.

When to use DB2 Admin Look Up special characters

Some fields support the Look Up function. You can use the question mark (?) Look Up character to search possible values that you can enter in the fields. If the question mark is entered in a field where the DB2 Admin Look Up function is not supported, an error message results. The question mark has its own unique meaning on table display panels.

The plus sign (+) Look Up character should only be used to select an object from the list returned by the DB2 Admin Look Up function. If the plus sign is entered on a table not provided by the DB2 Admin Look Up function, an invalid line command error message is returned.

Using search arguments to filter data on DB2 Admin panels

When you run queries to display information about DB2 objects or authorizations, you can use search arguments in certain input fields to filter the information that is displayed.

You can use a percent sign (%) or an asterisk (*) as a wildcard character in your search argument. If you use an asterisk as a wildcard character, DB2 Admin translates it to a percent sign. The asterisk is also displayed as a percent sign when the panel is re-displayed.

Lowercase characters in the search argument for Name, Owner, in D/L/H, Grantor, and Grantee are translated to uppercase characters unless you change the DB2 Admin default setting. If you change the value of the Capitalize object names parameter on the Change DB2 Admin Defaults panel (ADB2P2) to NO, lowercase characters will not be translated to uppercase characters where DB2 rules allow the

name to contain lowercase letters; lowercase characters cannot be translated to uppercase characters in database names, table space names, plan names, and package names that are not for trigger packages

For example, the panel in the following figure shows how you can use a search argument with wildcard characters in the Name field on the DB2 System Catalog panel (ADB21) to display all the databases in the DB2 system catalog with names that contain the characters 'DSN'.

```
 DB2 Admin -------------------- DB2X System Catalog --------------------- 17:34
 Option ===>

                                                                 More:     +
 Object options:                                         DB2 System: DB2X
   AO - Authorization options                            DB2 SQL ID: ISTJE
    G - Storage groups                P - Plans
    D - Databases                     L - Collections
    S - Table spaces                  K - Packages
    T - Tables, views, and aliases    M - DBRMs
    V - Views                         H - Schemas
    A - Aliases                       E - User defined data types
    Y - Synonyms                      F - Functions
    X - Indexes                       O - Stored procedures
    C - Columns                       J - Triggers
    N - Constraints                   Q - Sequences
   DS - Database structures          DSP - DS with plans and packages
  PDC - DB2 Pending definition changes  GV - Global variables
  XCU - Index cleanup

 Enter standard selection criteria (Using a LIKE operator, criteria not saved):
 Name  . . . . %DSN%                  > Grantor . . .          >
 Owner . . . .            >             Grantee . . .          >
 In D/L/H . .                    > Switch Catalog Copy . . . N   (N/S/C)
 And/or other selection criteria (option xC shows you columns for option x)
 Column  . . .                   > Operator . .       Value  . .
```

Figure 32. System Catalog (ADB21) – using search criteria

When you press Enter, DB2 Admin generates an SQL statement that searches the DB2 catalog using an SQL LIKE operator to qualify the search for the search criteria. The following figure shows the ISPF table display that DB2 Admin returns. All databases that meet the search criteria (have a name that contains the characters 'DSN') are displayed.

```
DB2 Admin ------------------ DB2X Databases -------------------- Row 1 of 25
Command ===>                                               Scroll ===> PAGE

Commands: GRANT  MIG  DIS  STA  STO  UTIL
Line commands:
 T - Tables  S - Table spaces  X - Indexes  G - Storage group  ICS - IC status
 DIS - Display database  STA - Start database  STO - Stop database  A - Auth
 ? - Show all line commands
                         Storage  Buffer           Created      Index
Select Name     Owner    Group    Pool    DBID By       T E BPool    I
       *        *        *        *         * *        * * *        *
------ -------- -------- -------- -------- ------ -------- - - -------- -
       ADBDSN   ADB      ADBGCH   BP1        271 ISTFL2     E BP2      Y
       DSNDB04  SYSIBM   SYSDEFLT BP1          4 SYSIBM       BP2      Y
       DSNDB06  SYSIBM                         6 SYSIBM     E BP0      Y
       DSNDB07  DSCGDB2  SYSDEFLT BP1          7 ISTJE    W   BP2      N
       DSNRGFDB DSCGDB2  SYSDEFLT BP1        257 ISTJE      E BP2      N
       DSNRLST  DSCGDB2  SYSDEFLT BP1        256 ISTJE      E BP2      N
       DSN8D81A DSCGDB2  DSN8G810 BP0        258 ISTJE      E BP2      N
       DSN8D81E DSCGDB2  DSN8G810 BP1        260 ISTJE    U   BP2      N
       DSN8D81P DSCGDB2  DSN8G810 BP0        259 ISTJE      E BP2      N
       DSN8D81U DSCGDB2  DSN8G81U BP1        261 ISTJE      E BP2      N
       GRGDSN01 DPGROTH  SYSDEFLT BP1        272 DPGROTH    E BP2      N
       GRGDSN02 DPGROTH  SYSDEFLT BP1        273 DPGROTH    E BP2      N
****************************** END OF DB2 DATA ********************************
```

Figure 33. System Catalog (ADB21) – list of qualifying databases

Sorting display data

You can sort alphabetically on one or more columns.

You can sort on any column by typing the SORT primary command followed by the column header name to be sorted. The keyboard shortcut for the column header name is the first letter of each word in the header name. You can also sort on any column by typing the SORT primary command, putting your cursor in the column to be sorted, and pressing Enter.

The following figure shows the information DB2 Admin returns when a SORT primary command is issued with the **CREATOR** parameter.

```
DB2 Admin ------------------- DB2X Databases -------------------- Row 1 of 25
Command ===>                                              Scroll ===> PAGE

Commands: GRANT  MIG  DIS  STA  STO  UTIL
Line commands:
 T - Tables  S - Table spaces  X - Indexes  G - Storage group  ICS - IC status
 DIS - Display database  STA - Start database  STO - Stop database  A - Auth
 ? - Show all line commands
                         Storage  Buffer          Created        Index
Select Name      Owner   Group    Pool     DBID  By       T E BPool    I
       *         *       *        *         * *  * *       * * *       *
------ -------- -------- -------- -------- ------ -------- - - -------- -
       ADBDCH   ADB      ADBGCH   BP1         271 ISTFL2     E BP2      Y
       DBEDB1   DPGROTH  SYSDEFLT BP1         272 DPGROTH    E BP2      Y
       DBEDB2   DPGROTH  SYSDEFLT BP1         273 DPGROTH    E BP2      Y
       DSNDB04  SYSIBM   SYSDEFLT BP1           4 SYSIBM       BP2      Y
       DSNDB06  SYSIBM                          6 SYSIBM     E BP0      N
       DSNDB07  DSCGDB2  SYSDEFLT BP1           7 ISTJE    W   BP2      N
       DSNRGFDB DSCGDB2  SYSDEFLT BP1         257 ISTJE      E BP2      N
       DSNRLST  DSCGDB2  SYSDEFLT BP1         256 ISTJE      E BP2      N
S      DSN8D81A DSCGDB2  DSN8G810 BP0         258 ISTJE      E BP2      Y
       DSN8D81E DSCGDB2  DSN8G810 BP1         260 ISTJE    U   BP2      Y
       DSN8D81P DSCGDB2  DSN8G810 BP0         259 ISTJE      E BP2      N
       DSN8D81U DSCGDB2  DSN8G81U BP1         261 ISTJE      E BP2      N
```

Figure 34. Databases after SORT CREATOR issued (ADB21D)

You can save a sort sequence for a panel so that the sequence is displayed every time that the panel is displayed (until specifically deleted). The saved sort sequence for each panel is saved in an ISPF table named ADBSORT in the user ISPPROF data set. You can specify the sort sequence for a panel by using the SORT command (without parameters). Panel ADBSORT is displayed, on which you can save or delete the sort sequence for that panel.

The following figure shows the information DB2 Admin returns when a SORT primary command is issued without parameters.

```
ADB2SORT  ------------------- DB2X Sort fields ------------- Row 1 to 15 of 21
Command ===>                                              Scroll ===> PAGE

Commands: SAVE  DELETE
Line commands: n - Sort col no  A - Ascending  D - Descending


                                               Table  Sort   Asc/
Select Column Name          Column Header Name  Col No Col no Desc
       *                    *                       *      * *
------ -------------------- ------------------- ------ ------ ----
       NAME                                          1      1 ASC
       CREATOR              OWNER                     2      0 ASC
       STGROUP              STORAGEGROUP              3      0 ASC
       BPOOL                BUFFERPOOL                4      0 ASC
       DBID                                           5      0 ASC
       IBMREQD                                        6      0 ASC
       CREATEDBY                                      7      0 ASC
       ROSHARE                                        8      0 ASC
       TIMESTAMP                                      9      0 ASC
       TYPE                 T                        10      0 ASC
       GROUP_MEMBER                                  11      0 ASC
       CREATEDTS                                     12      0 ASC
       ALTEREDTS                                     13      0 ASC
       ENCODING_SCHEME      E                        14      0 ASC
       SBCS_CCSID                                    15      0 ASC
```

Figure 35. Databases after SORT CREATOR issued (ADB21D)

Catalog navigation

You can navigate the catalog, which contains information about various DB2 objects. If you enter the S line command in the Select field next to database DSN8D81A in the panel in the previous figure, DB2 Admin displays all table spaces in database DSN8D81A. The results of issuing the S line command are shown in the following figure.

```
DB2 Admin ------------------ DB2X Table Spaces -------------------- Row 1 of 5
Command ===>                                              Scroll ===> PAGE

Commands: GRANT  MIG  DIS  STA  STO  ALL  DROP
Line commands:
 T - Tables  D - Database  A - Auth  G - Storage group  ICS - Image copy status
 DIS - Display database  STA - Start database  STO - Stop database
 ? - Show all line commands

Select Name     DB Name   Parts Bpool  L E S I C Tables  Act. pages  Segsz T L
       *        *          * *         * * * * *    *          *       * * *
------ -------- -------- ------ ------ - - - - - ------ ----------- ------ - -
DIS    DSN8S81D DSN8D81A      0 BP0    P N A N N      1          12      0  Y
       DSN8S81E DSN8D81A      4 BP0    P N A N N      1         120      0  Y
       DSN8S81R DSN8D81A      0 BP0    P N A N N      6           0      0  Y
       DSN8S81P DSN8D81A      0 BP0    R N A N N      4          24      4  Y
       DSN8S81S DSN8D81A      0 BP0    P N A N N      1           0      0  Y
```

Figure 36. Table spaces in a database (ADB21S)

You can issue commands against DB2 objects. From the Table Spaces panel, you can issue DB2 commands against DB2 objects. The previous figure demonstrates the use of the DIS line command against a DB2 table space. As shown in the following figure, output from a DB2 command is displayed in ISPF browse.

```
DB2 Admin ------- DB2X Browse DB2 Command Output ---  Line 00000000 Col 001 080
Command ===>                                               Scroll ===> PAGE

 -DIS DB(DSN8D81A) SPACENAM(DSN8S81D) LIMIT(*)


******************************** Top of Data **********************************
DSNT360I  DB2X ***********************************
DSNT361I  DB2X *  DISPLAY DATABASE SUMMARY
          *     GLOBAL
DSNT360I  DB2X ***********************************
DSNT362I  DB2X     DATABASE = DSN8D81A  STATUS = RW
                   DBD LENGTH = 16142
DSNT397I  DB2X
NAME     TYPE PART STATUS            PHYERRLO PHYERRHI CATALOG  PIECE
-------- ---- ---- ------------------ -------- -------- -------- -----
DSN8S81D TS        RW
******* DISPLAY OF DATABASE DSN8D81A ENDED      **********************
DSN9022I  DB2X DSNTDDIS 'DISPLAY DATABASE' NORMAL COMPLETION
***************************** Bottom of Data **********************************
```

Figure 37. ISPF Browse output after DB2 command (ADB2DB2O)

Refreshing data on DB2 Admin panels

As you work through DB2 Admin panels, you might want to refresh the original data on a panel after entering or changing data on that panel.

To refresh the data, enter the REFRESH primary command.

Using scrollable fields on DB2 Admin panels

To allow you to see the contents of input or output fields on panels that are not wide enough to display the entire contents of the field at once because the space on a DB2 Admin panel is limited, DB2 Admin uses ISPF scrollable fields.

The less than (<) and the greater than (>) symbols denote a scrollable field. A > symbol indicates that the field can be scrolled to the right, and a < symbol indicates that the field can be scrolled to the left. Both symbols are displayed when you are in the middle of data and can scroll either left or right. You can use the following ISPF commands to work with the field:

- To scroll through the field, type LEFT or RIGHT in the command field, position the cursor in the field, and press Enter.
- To see the entire contents of the field at once, type EXPAND in the command field, position your cursor in the scrollable field, and press Enter.
- To clear the contents of the field, type ZCLRSFLD in the command field, position your cursor in the scrollable field, and press Enter. (If your level of z/OS does not support the ZCLRSFLD command, you can use the EXPAND command to display the entire contents of the field, and then clear the contents of the field in the pop-up window.)

Tip: You can assign your PF keys to be the LEFT, RIGHT, EXPAND, and ZCLRSFLD commands. Using a PF key simulates both typing in the command and pressing Enter.

For example, the following figure shows the ALTER Table panel. On this panel, New schema and New name are scrollable input fields. Old schema and Old name are scrollable output fields. Column Name is a scrollable input/output column.

```
 ADB27C in ------------------ DSNA ALTER Table --------------- Row 1 to 4 of 4
  Command ===>                                         Scroll ===> CSR

  New schema . . MARLINO   >               Old schema: MARLINO
  New name . . . PJRI              >       Old name  : PJRI
  Partitions . : 0                         New DB  . . PJDBRI2
  Rows per page: 145                       New TS  . . PJTSRI2

  Commands:  NEXT  CONSTRAINTS  TBLOPTS         HASH
  Line commands:
   I - Insert  U - Update  D - Delete  R - Repeat  LAB - Label  COM - Comment
   M - Move    A - After   B - Before  X - Index   RES - Reset update
   UM - Update XML modifiers
                                                       Old Operation
  Sel Column Name        Col No Col Type     Length  Scale N D Col No Type
      *                       * *                 *      * * *      * *
  --- ----------------->  ------ -------- ----------- ------ - - ------ ---------

      P1                      1 INTEGER         4       0 N N     1
      P2                      2 INTEGER         4       0 N N     2
      P3                      3 INTEGER         4       0 N N     3
      P4                      4 INTEGER         4       0 N N     4
  ****************************** END OF DB2 DATA ******************************
```

Figure 38. Scrollable fields on the ALTER Table panel

Checking the status of DB2 Admin

On any DB2 Admin panel, you can check the status of DB2 Admin by using the STATUS primary command.

When you use the STATUS command, the DB2 Admin Status panel is displayed, as shown in the following figure.

Using the DB2 Admin Status panel, you can verify the environment in which DB2 Admin is running (for example, the current SQL ID and the DB2 release). In addition, execution control counts are displayed.

```
DB2 Admin ----------------- DB2X DB2 Admin Status --------------------- 11:07
Option ===>

Current DB2 Admin status: Accessing the local system
                                                            More:    +
Local DB2 subsystem name: DB2X
Userid              : ISTJE
Current SQL ID      : ISTJE

DB2 release         : 810
DB2 product         : DB2

Catalog qualifier   : SYSIBM - running directly on catalog tables
DDF location        : (blank) - running locally
Current server      : CPHMVS1_DB2X - local server
Remote subsystem name : n/a

Execution totals           Counts                                Counts
Prepare             :         4   Execute dynamically   :            0
Describe            :         6   - Set                 :            0
Open                :         4   - Insert              :            0
Fetch               :      1039   - Update              :            0
Close               :         4   - Delete              :            0
Commit              :         4   - Create              :            0
Rollback            :         0   - Drop                :            0
Connect             :         0   - Alter               :            0
Set                 :         2   - Comment             :            0
User rows affected  :         0   - Label               :            0
                                  - Grant               :            0
                                  - Revoke              :            0
                                  - Rename              :            0
                                  - Commit              :            0
                                  - Rollback            :            0
                                  - Other dynamic       :            0
Use the RESET command to reset the counts
```

Figure 39. The DB2 Admin Status panel (ADB2STAT)

DB2 Administration Menu panel

The DB2 Administration Menu panel is the main menu for accessing DB2 Admin functions.

The DB2 Administration Menu panel, as shown in the following figure, is referred to throughout this information.

Attention: You can use the TSO split screen to access the DB2 Admin Tool. However, if the DB2 systems you are accessing are different version levels, you might experience unexpected problems such as a system abend 0C4, ABEND0C4. To avoid problems, ensure that the different DB2 systems are the same version level.

DB2 Admin includes a sample application as part of the product. You can access the sample application from this panel by specifying the "hidden" option S.

```
DB2 Admin -------------- DB2 Administration Menu 11.2.0 ------------------ 00:49
Option ===> 1

   1 - DB2 system catalog                              DB2 System: DB2X
   2 - Execute SQL statements                          DB2 SQL ID: ISTJE
   3 - DB2 performance queries                         Userid    : ISTJE
   4 - Change current SQL ID                           DB2 Schema: ISTJE
   5 - Utility generation using LISTDEFs and TEMPLATEs DB2 Rel   : 1115
   P - Change DB2 Admin parameters
  DD - Distributed DB2 systems
   E - Explain
   Z - DB2 system administration
  SM - Space management functions
   W - Manage work statement lists
   X - Exit DB2 Admin
  CC - DB2 catalog copy version maintenance
  CM - Change management

Interface to other DB2 products and offerings:
   I  DB2I   DB2 Interactive
   C  DB2 Object Comparison Tool
```

Figure 40. DB2 Administration Menu (ADB2)

DB2 SYSTEM CATALOG
Select this option to display information from the catalog about DB2 objects and/or authorizations for those objects.

EXECUTE SQL STATEMENTS
Select this option to execute SQL statements.

DB2 PERFORMANCE QUERIES
Select this option to run performance and space utilization queries.

CHANGE CURRENT SQL ID
Select this option to change your current SQL ID. This is the same as issuing the DB2 Admin primary command SQLID.

UTILITY GENERATION USING LISTDEFS AND TEMPLATES
Choose this option to generate utilities using LISTDEFs and TEMPLATEs.

CHANGE DB2 ADMIN PARAMETERS
Select this option to change DB2 Admin parameters.

DISTRIBUTED DB2 SYSTEMS
Select this option to see the system catalog panels for a remote DB2 system.

EXPLAIN
Select this option to:
- Enter an SQL statement and see the resulting rows in a plan table (PLAN_TABLE).
- List rows from a plan table and see how DB2 will execute SQL statements in application plans, or packages that were bound with EXPLAIN(YES).
- Create and upgrade a plan table.

DB2 SYSTEM ADMINISTRATION
Select this option to display a list of system administration functions.

SPACE MANAGEMENT FUNCTIONS
Select this option to perform space manager functions.

MANAGE WORK STATEMENT LISTS
Select this option to display the work statement list library and to manage work statement lists.

DB2 CATALOG COPY VERSION MAINTENANCE
Select this option to maintain and update the Catalog Copy Version Table. This option appears only if you customized your system for support of multiple catalog copies.

CHANGE MANAGEMENT
Select this option to use the Change Management functions. You can manage objects such as changes, versions, masks, and ignores. You can also complete tasks such as managing report changes. This option is displayed only if DB2 Admin has been customized such that the use of Change Management is enabled.

Setting panel display options

You can customize settings across all of the DB2 Admin panels that display lists of objects by using the OPTIONS primary command.

When you issue the OPTIONS D command, the Panel Display Options pop-up panel is displayed. Using the Panel Display Options panel, you can select which of the following items you want to be included in the panel:

- DB2 Admin action bar
- Panel instructions
- Primary commands
- Line commands
- Filter line

Select an item by typing a slash (/) character in front of the item, then press the Enter key. Deselecting some of the items will result in a simpler-looking panel that displays more data objects.

Note: When the action bar is selected, the DB2 Admin Tool action bar will be shown on select panels. Not all panels support this feature.

Chapter 6. DB2 Admin tutorial

The topics in this information demonstrate how to navigate DB2 Admin and introduce you to some of its major functions.

Comprehensive information about all of DB2 Admin functionality is contained in Part 3: Using DB2 Admin.

Remember: This tutorial is based on the DB2 sample database that is provided with DB2. If you do not have the sample database installed on your system, you can still follow along with the tutorial by using one of your own databases.

Topics:
- "Running queries"
- "Sorting display data" on page 136
- "Running utilities" on page 145
- "Granting authorizations" on page 148
- "Binding plans and packages" on page 149
- "Displaying detailed information about an object" on page 153
- "Reverse engineering objects" on page 154

Figure 41 on page 144 shows the DB2 Administration Menu panel that is displayed when you start DB2 Admin. The top of the panel shows the DB2 Admin functions you can choose. The release level and mode of your DB2 subsystem affect the options, within the functions, that are available to you. The bottom of the panel shows other DB2 tools (in this case, DB2 Interactive and DB2 Object Comparison Tool) that can be invoked from the main menu; this is a customization option.

Running queries

You run queries to display and filter information about database objects.

Choose option 1 on the panel, as shown in the following figure, to display the DB2 System Catalog panel.

```
DB2 Admin -------------- DB2 Administration Menu 11.2.0 ------------------ 00:49
Option ===> 1

   1 - DB2 system catalog                             DB2 System: DB2X
   2 - Execute SQL statements                         DB2 SQL ID: ISTJE
   3 - DB2 performance queries                        Userid   : ISTJE
   4 - Change current SQL ID                          DB2 Schema: ISTJE
   5 - Utility generation using LISTDEFs and TEMPLATEs DB2 Rel   : 1115
   P - Change DB2 Admin parameters
  DD - Distributed DB2 systems
   E - Explain
   Z - DB2 system administration
  SM - Space management functions
   W - Manage work statement lists
   X - Exit DB2 Admin
  CC - DB2 catalog copy version maintenance
  CM - Change management

Interface to other DB2 products and offerings:
   I  DB2I   DB2 Interactive
   C  DB2 Object Comparison Tool
```

Figure 41. DB2 Administration Menu (ADB2)

Choose option D on this panel, which displays the databases in the DB2 system
catalog. You can filter the databases that are displayed by specifying a search
argument in the **Name** field. You can use a percent sign (%) or an asterisk (*) as a
wildcard character in your search argument.

```
ADB21 min --------------------- DSNB System Catalog --------------------- 16:17
 Option ===> D


 Object options:                                       DB2 System: DSNB
   AO - Authorization options                          DB2 SQL ID: SMITH
    G - Storage groups                  P - Plans
    D - Databases                       L - Collections
    S - Table spaces                    K - Packages
    T - Tables, views, and aliases
    V - Views                           H - Schemas
    A - Aliases for tables and views    E - User defined data types
    Y - Synonyms                        F - Functions
    X - Indexes                         O - Stored procedures
    C - Columns                         J - Triggers
    N - Constraints                     Q - Sequences and aliases
   DS - Database structures            DSP - DS with plans and packages
  PDC - DB2 pending definition changes  GV - Global variables
  XCU - Index cleanup

 Enter standard selection criteria.  Settings: LIKE operator;  Criteria not save
 Name  . . . . DB*              > Grantor . . .         >
 Owner . . . .           >        Grantee . . .         >
 In D/L/H  . .                  > Switch Catalog Copy . . . N   (N/S/C)
 And/or other selection criteria (option xC shows you columns for option x)
 Column  . . .                  > Operator . .      Value . .
```

Figure 42. System Catalog (ADB21) – object options

The following figure shows the ISPF table display panel that DB2 Admin returns.
All databases that meet the search criteria are displayed in the **Name** field.

```
  ADB21D in ------------------ DSNB Databases --------------------------------

  Commands: GRANT  MIG  DIS  STA  STO  UTIL  CT
  Line commands:
   T - Tables  S - Table spaces  X - Indexes  G - Storage group  ICS - IC status
   DIS - Display database  STA - Start database  STO - Stop database  A - Auth
   ? - Show all line commands
                           Storage  Buffer            Created       Index
  Select Name     Owner    Group    Pool     DBID By          T E BPool    I
         *        *        *        *          * *           * * *        *
  ------ -------- -------- -------- -------- ------ -------- - - -------- -
         DB1      ADB      ADBGCH   BP1         271 ISTFL2     E BP2      Y
         DB1A     DPGROTH  SYSDEFLT BP1         272 DPGROTH    E BP2      Y
         DB12     DPGROTH  SYSDEFLT BP1         273 DPGROTH    E BP2      Y
         DB14     SYSIBM   SYSDEFLT BP1           4 SYSIBM       BP2      N
         DB16     SYSIBM                          6 SYSIBM     E BP0      N
         DB17     DSCGDB2  SYSDEFLT BP1           7 ISTJE      W BP2      N
         DB1B     DSCGDB2  SYSDEFLT BP1         257 ISTJE      E BP2      N
  ****************************** END OF DB2 DATA ******************************
```
Figure 43. System catalog databases (ADB21D)

Running utilities

You can run DB2 utilities from DB2 Admin.

Redisplay the Table Spaces panel. Specify line command UTL for table space
DSN8S81D. DB2 Admin responds by displaying the utilities that can be run against
a table space, as shown in the following figure.

```
ADB2US in ----------------- DSNB Table Space Utilities ----------------- 23:16
Option ===>

 Execute utility on                                        DB2 System: DB2X
    table space DSN8D81A.DSN8S81D                          DB2 SQL ID: ISTJE
                                                                    More:    +
   C - Copy full            CI - Copy incremental   C2 - Copytocopy
  CC - Copy concurrent
   E - Mergecopy            EN - Mergecopy newcopy
   K - Check index          KD - Check data         KL - Check LOB
  LC - Load with Cross loader
   M - Modify               NW - Repair Auxwarn     NX - Repair Auxcheckpend
   N - Repair nocopypend    NA - Repair nocheckpend NB - Repair norcvrpend
  NC - Repair catalog       NL - Repair Levelid     NR - Repair noreorgpend
   O - Reorg                OU - Reorg unload only  OO - Online reorg
  OC - Reorg with Inline Copy
   P - Report recovery       Q - Quiesce
   R - Runstats             RT - Runstats table all RR - Runstats report
  RX
   V - Recover              VC - Recover tocopy     VG - Recover to last GDG
  VI - Rebuild index        VR - Recover torba      VL - Recover logonly
  DG - Define GDG for copy data sets                VP - Recover tologpoint
   U - Unload


  SM - Standard Maintenance  C O R
  BP - Change batch job parameters
  TU - Specify Template Usage

Utility control options
  Review/change options  . . . . . NO   (Yes/No)
  Generate work statement list . . NO   (Yes/No)
  Generate template statements . . NO   (Yes/No)
  Generate modify after copy . . . NO   (Yes/No)
```

Figure 44. Table Space Utilities menu (ADB2US)

Note: The LC option is displayed only in the following situations:

- The table does not contain XML columns
- The panel is displayed for one table space
- The table space contains only one table
- The table space is not an LOB table space

You can run the COPY utility against the table space by specifying option C, which
requests a full image copy. The following figure shows the JCL that DB2 Admin
returns to you. The JCL is ready to be submitted.

```
   File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
------------------------------------------------------------------------------
EDIT       ISTJE.SPFTEMP2.CNTL                          Columns 00001 00072
Command ===>                                            Scroll ===> PAGE
000007 //*
000008 //***********************************************************************
000009 //*
000010 //* DB2 ADMIN GENERATED JOB TO RUN COPY ON SELECTED TABLESPACES
000011 //*
000012 //***********************************************************ADB2USC***
000013 //*
000014 //***********************************************************************
000015 //* STEP COPY: COPY TABLESPACE DSN8D81A.DSN8S81D
000016 //***********************************************************ADB2USC1**
000017 //COPY EXEC DSNUPROC,SYSTEM=DB2X,
000018 //              LIB='SYS1.DSNDB2X.SDSNLOAD',
000019 //              UID='ISTJE'
000020 //DSNUPROC.SYSCOPY DD DSN=ISTJE.DB2X.IC.DSN8D81A.DSN8S81D(+1),
000021 //              DISP=(NEW,CATLG),
000022 //              SPACE=(8192,(7,5),RLSE),
000023 //              UNIT=SYSDA
000024 //DSNUPROC.SYSIN  DD  *
000025 COPY TABLESPACE DSN8D81A.DSN8S81D DSNUM ALL FULL YES
000026 /*
000027 //***********************************************************************
000028 //* STEP MOD: MODIFY RECOVERY TABLESPACE DSN8D81A.DSN8S81D
000029 //***********************************************************************
000030 //MOD EXEC DSNUPROC,SYSTEM=DB2X,
000031 //              LIB='SYS1.DSNDB2X.SDSNLOAD',
000032 //              UID='ISTJE'
000033 //DSNUPROC.SYSIN  DD  *
000034 MODIFY RECOVERY TABLESPACE DSN8D81A.DSN8S81D DSNUM ALL
000035     DELETE AGE(35)
000036 /*
****** **************************** Bottom of Data ****************************
```

Figure 45. JCL for a utility

Back to the Table Spaces panel again, you can determine what tables are in a table space by issuing the T line command. The following figure shows the tables in table space DSN8S81D.

```
DB2 Admin ----------- DB2X Tables, Views, and Aliases ----------- - Row 1 of 1
Command ===>                                            Scroll ===> PAGE

Commands: GRANT  MIG  ALL  CT
Line commands:
 C - Columns  A - Auth  L - List  X - Indexes  S - Table space  D - Database
 V - Views  T - Tables  P - Plans  Y - Synonyms  SEL - Select prototyping
 ? - Show all line commands

Sel   Name              Owner    T DB Name  TS Name   Cols      Rows Checks
      *                 *        * *        *         *           *      *
----- ----------------- -------- - -------- -------- ------ ---------- ------
      DEPT              DSN8810  T DSN8D81A DSN8S81D     5         14      0
      VDEPT             DSN8810  V DSN8D81A DSN8S81D     4         -1      0
****************************** END OF DB2 DATA ******************************
```

Figure 46. Tables in a table space (ADB21T)

To see the columns in a table, issue the C line command against the DEPT table. The result is shown in the following figure.

```
DB2 Admin ----- DB2X Columns in Table: DSN8810.DEPT -------------- Row 1 of 5
Command ===>                                              Scroll ===> PAGE

Line commands:
 T - Tables  X - Indexes  A - Auth  GR - Grant  H - Homonyms  I - Interpret
 UR - Update runstats  LAB - Label  COM - Comment  DI - Distribution stats
 PST - Partition stats   E  - Source data type  SEQ - Identity column info
 ? - Show all line commands

Select Column Name       Col No Col Type Length Scale  Null Def FP   Col Card
       *                     *  *             *     *  *   *   *            *
------ ----------------- ------ -------- ------ ------ ---- --- -- -----------
       DEPTNO                1 CHAR          3      0 N    N   N           14
       DEPTNAME              2 VARCHAR      36      0 N    N   N           -1
       MGRNO                 3 CHAR          6      0 Y    Y   N            9
       ADMRDEPT              4 CHAR          3      0 N    N   N            3
       LOCATION              5 CHAR         16      0 Y    Y   N           -1
****************************** END OF DB2 DATA ******************************
```

Figure 47. Columns in a table (ADB21TC)

To see the indexes for a table, issue the X line command against the DEPT table. The
following figure shows the information that is returned.

```
DB2 Admin --------------------- DB2X Indexes --------------------- Row 1 of 3
Command ===>                                              Scroll ===> PAGE

Commands: DIS  STA  STO
Line commands:
 T - Tables  D - Database  G - Storage group  P - Plans  C - Columns
 DIS - Display database  STA - Start database  STO - Stop database
 ? - Show all line commands

                      Index                    Table         C C C
Select Index Name     Owner    Table Name      Owner    U  Cols G D L T
       *              *        *               *        *   * * * * *
------ -------------- -------- --------------- -------- - ------ - - - -
       XDEPT1         DSN8810  DEPT            DSN8810  P    1 N Y N 2
       XDEPT2         DSN8810  DEPT            DSN8810  D    1 N Y N 2
       XDEPT3         DSN8810  DEPT            DSN8810  D    1 N Y N 2
****************************** END OF DB2 DATA ******************************
```

Figure 48. Indexes for a table (ADB21X)

Granting authorizations

You can grant authorizations with DB2 Admin.

You can find the authorizations for any DB2 object by issuing the A line command.
The following figure shows the output that DB2 Admin returns when the A line
command is issued against table DEPT.

```
 DB2 Admin --------------- DB2X Table Authorizations -------------- Row 1 of 2
 Command ===>                                                 Scroll ===> PAGE

 Commands: REVOKE  GRANT
 Line commands:
  R - Revoke  GR - Grant  T - Table  I - Interpretation   U   D   I S U   R
   CA - Column authorizations                             P A E I N E P R E
                                                          D L L N S L D E F T
                                                          C T E D E E A F C R
                       G                          H Date  O E T E R C T E O I
 S  Grantor  Grantee  T Schema    Name            G Grant  L R E X T T E R L G
    *        *        * *         *               * *      * * * * * * * * * *
 -- -------- -------- - -------- ---------------- - ------ - - - - - - - - - -
 GR DSN8810  DSN8810    DSN8810  DEPT             S 010524  G G G G G G G   G
    DSCGDB2  PUBLIC*    DSN8810  DEPT             S 010524    Y   Y Y Y
 ***************************** END OF DB2 DATA *****************************
```

Figure 49. Authorizations for a DB2 object (ADB2AT)

Issue the GR line command to grant privileges for the object. The following figure
shows the information that DB2 Admin returns when GR is issued against table
DEPT.

```
 DB2 Admin ----------------- DB2X Grant Table Privileges ----------------- 00:53
 Command ===>

  GRANT

  Specify Y or G (for WITH GRANT OPTION) or ' ' (for none)

   ALL            INDEX          UPDATE
   ALTER          INSERT         REFERENCES
   DELETE         SELECT         TRIGGERS

  ON TABLE

  OWNER . . . . VNDEJB    >
  TABLE . . . . ERICTB1             >

  TO

  To  . . . . . USERX
                                                                            >
```

Figure 50. Grant privileges for a table (ADB2GT)

Binding plans and packages

You can bind plans and packages in DB2 Admin.

From the System Catalog menu, select option P to list the application plans in the
catalog as shown in the following figure.

```
DB2 Admin ---------------- DB2X Application Plans ---------------- Row 1 of 25

Commands: BIND  REBIND  FREE  GRANT
Line commands:
 DP - Depend  A - Auth  T - Tables  V - Views  X - Indexes  S - Table spaces
 Y - Synonyms  M - DBRMs  RB - Rebind  F - Free  B - Bind  GR - Grant
 PL - Package list  LP - List PLAN_TABLE  I - Interpret  ENDI - Enab/disab con
 K - Local packages  SQ - SQL  D - Databases  RO - Role
                        Bind   Bind   V I V O Bound    Quali-    Pack A R E D
Select Name     Owner   Date   Time   D S A P By       fier      Lists Q L X R
       *        *       *      *      * * * * *        *               * * * * *
------ -------- ------->  ------ ------ - - - - -------- ------->  ------ - - - -
       ADBTEP2  DSCGDB2  010828 100153 B S Y Y ISTFL2   DSCGDB2        1 U C N
       ADBV3    DSCGDB2  010912 024459 B S Y Y ISTFL    DSCGDB2        2 U C Y
       ADB2GEN  DSCGDB2  010623 005531 B S Y Y ISTJE    DSCGDB2        1 U C Y
       ADB2GE2  DSCGDB2  010526 003803 B S Y Y ISTFL    DSCGDB2        1 U C Y
       ADB21    DSCGDB2  010623 004026 B S Y Y ISTJE    DSCGDB2        1 U C N
       ADB31    DSCGDB2  011030 170150 B S Y Y ISTJE    DSCGDB2        1 U C N
       DB2E81   DPGROTH  011029 145636 R S Y Y DPGROTH  DPGROTH        0 U C Y
       DSNEDCL  DSCGDB2  010524 190326 R S Y Y ISTJE    DSCGDB2        1 U C N
       DSNESPCS DSCGDB2  010524 190324 R S Y Y ISTJE    DSCGDB2        1 U C N
       DSNESPRR DSCGDB2  010524 190325 R R Y Y ISTJE    DSCGDB2        1 U C N
M      DSNTIAD  DSCGDB2  010524 024119 R S Y Y ISTJE    DSCGDB2        0 U C N
****************************** END OF DB2 DATA ******************************
```

Figure 51. Application plans (ADB21P)

Use the M line command from the Application Plans panel to display the DBRMs
for an application plan. The following figure shows the output that DB2 Admin
returns when the M line command is issued against application plan DSNTIAD.

```
DB2 Admin --------------------- DB2X DBRMs --------------------- Row 1 of 1
Command ===>                                                Scroll ===> PAGE

Line commands:
 P - Plans  B - Browse DBRM  S - SQL statements  I - Interpretation

S Name      Owner    PL Name  Q C H P Date P Time    PDS Name
  *         *        *        * * * *  *        *
- -------- ------->  -------- - - - - ------ -------- ----------------------------
S DSNTIAD  DSCGDB2   DSNTIAD  N N B 010524 02410439 DB2.DSN810.DBRMLIB.DATA
****************************** END OF DB2 DATA ******************************
```

Figure 52. DBRMs for an application plan (ADB21M)

To request the actual SQL statements in the DBRM, issue line command S. The
result is shown in the following figure.

```
  DB2 Admin ---------------- Extracted SQL ---------------- Columns 00001 00072
  Command ===>                                              Scroll ===> PAGE

  ****** ***************************** Top of Data ******************************
  000001 -- SQL statements in DBRM: DSNTIAD.DSNTIAD
  000002 -- SQL in stmt: 982
  000003 WHENEVER SQLERROR GO TO EXECERR
  000004 -- SQL in stmt: 983
  000005 WHENEVER SQLWARNING GO TO EXECWRN
  000006 -- SQL in stmt: 984
  000007 WHENEVER NOT FOUND GO TO EXECWRN
  000008 -- SQL in stmt: 1226
  000009 CONNECT
  000010 -- SQL in stmt: 1278
  000011 CONNECT RESET
  000012 -- SQL in stmt: 1405
  000013 CONNECT TO :H
  000014 -- SQL in stmt: 1528
  000015 SET CONNECTION :H
  000016 -- SQL in stmt: 1649
  000017 RELEASE CURRENT
  000018 -- SQL in stmt: 1700
  000019 RELEASE ALL
  000020 -- SQL in stmt: 1780
  000021 RELEASE ALL PRIVATE
  000022 -- SQL in stmt: 1829
  000023 RELEASE ALL SQL
  000024 -- SQL in stmt: 1938
  000025 RELEASE :H
  000026 -- SQL in stmt: 1993
  000027 EXECUTE IMMEDIATE :H
  ****** **************************** Bottom of Data ****************************
```

Figure 53. SQL statements in a DBRM (ADB21KSE)

From the Application Plans panel, you can issue a Bind, Rebind, or Free line
command for a particular plan. You can also issue a BIND, REBIND, FREE, or
GRANT primary command for all plans listed.

The following figure shows the result when you request a Bind of application plan
DSNTIAD.

```
ADB21PB n ----------------- DBAB Bind Application Plan ----------------- 13:41
Command ===>

                                                                More:     +
 Verify BIND parameters:

 BIND PLAN(
 Plan name  . . . . . DSNESPRR
 OWNER  . . . . . . . DB2ADM   >
 QUALIFIER  . . . . . DB2ADM   >  (qualifier to resolve unqualified SQL)
 PKLIST . . . . . . . *.DSNESPRR.DSNESM68 *.DSNTIAP.DSNTIAP        >
 DEFER(PREPARE) . . . NO      (Yes/No, used for distributed dynamic SQL)
 VALIDATE . . . . . . R       (Run or Bind, Bind preferred)
 ISOLATION  . . . . . RR      (CS, RR, RS, or UR)
 CACHE  . . . . . . . 3072    (cache size in bytes for authorization IDs)
 ACQUIRE  . . . . . . U       (Use or Allocate, Use preferred)
 RELEASE  . . . . . . C       (Commit or Deallocate, Commit preferred)
 EXPLAIN  . . . . . . NO      (Yes/No, to explain access path)
 CURRENTDATA  . . . . NO      (Yes/No)
 CURRENT SERVER . . .             >  (blank=local, else first location)
 ACTION . . . . . . . REPLACE (Add or Replace)
 RETAIN . . . . . . . YES     (Yes/No)  (Retain auth list)
 ENABLE . . . . . . .         (use ? to get current values from the catalog)
 DISABLE  . . . . . .         (use ? to get current values from the catalog)
 En/disable names . .
                             (use ? to get current values from the catalog)
 DEGREE . . . . . . . 1       (1 or ANY)  (Parallelism)
 SQLRULES . . . . . . D       (DB2 or STD)
 DISCONNECT . . . . . E       (Explicit, Automatic, or Conditional)
 DYNAMICRULES . . . .         (Run or Bind)
```

Figure 54. A BIND of an application plan (ADB21PB) (1 of 2)

```
 ENABLE . . . . . . .         (use ? to get current values from the catalog)
 DISABLE  . . . . . .         (use ? to get current values from the catalog)
 En/disable names . .
                             (use ? to get current values from the catalog)
 DEGREE . . . . . . . 1       (1 or ANY)  (Parallelism)
 SQLRULES . . . . . . D       (DB2 or STD)
 DISCONNECT . . . . . E       (Explicit, Automatic, or Conditional)
 DYNAMICRULES . . . .         (Run or Bind)
 KEEPDYNAMIC  . . . . NO      (Yes/No)
 REOPT(VAR) . . . . . NONE    (N - None, Y - Always, 1 - Once, or A-Auto)

 OPTHINT  . . . . . .         >
 PATH . . . . . . . .                                          >
 ENCODING . . . . . . 37      (ASCII, EBCDIC, UNICODE or ccsid)
 IMMEDWRITE . . . . . NO      (Yes,No or PH1)
 ROUNDING . . . . . . HALFEVEN (Ceiling, Down, Floor, HalfDown,)
                              (HalfEven, HalfUp, or Up)
 CONCURRENTACCESSRES          (U - Usecurrentlycommitted or)
                              (W - Waitforoutcome)
```

Figure 55. A BIND of an application plan (ADB21PB) (2 of 2)

DB2 Admin uses the catalog to automatically find the DBRM members and
libraries for the bind. These are displayed when you press Enter, as shown in the
following figure.

```
ADB21PB n ----------------- DBAB Bind Application Plan ---------------- 13:41
Command ===>

                                                             More:     +
 Verify BIND parameters:

 BIND PLAN(
 Plan name  . . . . . DSNTIAD
 OWNER  . . . . . . . DSCGDB2    >
 QUALIFIER  . . . . . DSCGDB2    >  (qualifier to resolve unqualified SQL)
 PKLIST . . . . . . .            >
```

Figure 56. DBRM members and libraries for the BIND (ADB21PB)

If an SQL error occurs, DB2 Admin displays the DSNTIAR message, as shown in
the following figure.

```
DB2 Admin -------------------- DB2 Error Display 1 -------------------- 12:54
Command ===>
Rollback done
   SQLCODE : -206                      DSNTIAR CODE :  0

DSNT408I SQLCODE = -206, ERROR:  T.TYP IS NOT A COLUMN OF AN INSERTED TABLE,
         UPDATED TABLE, OR ANY TABLE IDENTIFIED IN A FROM CLAUSE, OR IS NOT A
         COLUMN OF THE TRIGGERING TABLE OF A TRIGGER
DSNT418I SQLSTATE   = 42703 SQLSTATE RETURN CODE
DSNT415I SQLERRP    = DSNXORSO SQL PROCEDURE DETECTING ERROR
DSNT416I SQLERRD    = -600  0  0  -1  0  0 SQL DIAGNOSTIC INFORMATION
DSNT416I SQLERRD    = X'FFFFFDA8'  X'00000000'  X'00000000'  X'FFFFFFFF'
         X'00000000'  X'00000000' SQL DIAGNOSTIC INFORMATION
```

Figure 57. DSNTIAR error messages

When you press Enter, a second error panel opens to display the actual SQL
statement that caused the error as shown in the following figure.

```
DB2 Admin -------------------- DB2 Error Display 2 -------------------- 12:54
Command ===>

   SQLCODE : -206                      DSNTIAR CODE :  0

PREPARE

SELECT T.* FROM SYSIBM.SYSTABLES T  WHERE  T.CREATOR LIKE 'DSN and T.TYP = 'V'
 FOR FETCH ONLY
```

Figure 58. SQL statement in error

Displaying detailed information about an object

You can display detailed information about an object in DB2 Admin.

If you want interpretive information about an object in the DB2 catalog, you can
use the I line command on the Application Plans panel. The following figure
shows the result when you issue the I line command against application plan
DSNTIAD.

```
ADB21PI1  --------- DSNB Interpretation of an Object in SYSPLAN --------- 11:46
Command ===>

Details for application plan : DSNTIAD
                                                                 More:      +
Authorization ID of owner  . . . : DB2ADM
Authorization ID of creator  . . : ISTJE
Creator type . . . . . . . . . . : Auth ID
Qualifier for unqualified SQL  . : DSCGDB2
Date of latest BIND of plan  . . : 040524   (yymmdd)
Time of latest BIND of plan  . . : 02411994 (hhmmssth)
Time when the plan was bound . . : 2004-05-24-02.41.19.948290
Version under which plan bound . : P - DB2 V11
SQL rules specified at BIND  . . : D - DB2
Cache size for auth IDs in bytes : 1024
Operative status of plan . . . . : Plan is valid and operative
Resource and authorization check : At plan allocation time
Plan base section size (bytes) . : 2632     (in EDM pool during execution)
Average DML section size (bytes) : 0        (loaded when needed during exec)
Plan bound with EXPLAIN option . : NO
Plan bound with DEFER(PREPARE) . : No - DEFER(PREPARE) not specified
Number of PACKAGE list entries . : 0
Number of enabled/disabled sys . : 0
Current server . . . . . . . . . :
Disconnect option used . . . . . : E - explicit. Release locations at commit
Data concurrency . . . . . . . . : C - required for ambiguous cursors
  Effect on blocking . . . . . . : Inhibit blocking for ambiguous cursors
DEGREE of I/O parallelism  . . . : 1 - parallel I/O inhibited
Group member that performed BIND :
Dynamic SQL rules  . . . . . . . : Not specified - use the rules for the plan
Re-optimize SQL at execution time: No - access path determined at BIND time
Keep prepared dynamic SQL stmts  : No - are destroyed at COMMIT
Protocol for 3 part names  . . . : D
Function resolved at . . . . . . : 2004-05-24-02.41.19.894713
Optimizer hint identifier  . . . :
Encode CCSID . . . . . . . . . . : 277
Write group buffer pool pages  . : Normal write
Catalog table uncoding scheme  . : Unicode
SQL path for resolving UDT,UDF,SP:

ROUNDING option used on last bind: Created prior to V9
Concurrent access  . . . . . . . : Not specified - inherit from DB2 ZPARM

Resource allocation information :
Resources acquired . . . . . . . : When first used
Resources released . . . . . . . : At COMMIT
Isolation level  . . . . . . . . : Cursor stability
```

Figure 59. Interpretation of an object (ADB21PI1)

Reverse engineering objects

You can reverse engineer objects in your DB2 catalog (that is, extract the DDL that
is required to re-create the DB2 objects).

The starting point for reverse engineering can be databases, table spaces, tables,
aliases, synonyms, schemas, data types, functions, stored procedures, triggers,
sequences, or storage groups.

The following figure shows the panel that is displayed when the GEN line
command is issued to reverse engineer the DSN8D81A database:

```
ADB2GEN n -------------- DB2X Generate SQL from DB2 catalog ------------- 11:34
Option ===>

Generate SQL statements for database DSN8D81A          DB2 System: DB2X
                                                       DB2 SQL ID: JSMITH


SQL statement types to be generated from the DB2 catalog:
  CREATE DATABASE  . . . . Y (Y,N)    GRANT access ON DATABASE . . Y (Y,N,A,R)
  CREATE TABLESPACE  . . . Y (Y,N)    GRANT access ON TABLESPACE . Y (Y,N,A,R)
  CREATE TABLE . . . . . . Y (Y,N)    GRANT access ON TABLE  . . . Y (Y,N,A,R)
  CREATE VIEW  . . . . . . Y (Y,N,D)  GRANT access ON VIEW . . . . Y (Y,N,A,R)
  CREATE INDEX . . . . . . Y (Y,N)    ALTER TABLE ADD FOREIGN KEY. Y (Y,N,D)
  CREATE SYNONYM . . . . . Y (Y,N)    LABEL ON . . . . . . . . . . Y (Y,N)
  CREATE ALIAS . . . . . . Y (Y,N)    COMMENT ON . . . . . . . . . Y (Y,N)
  CREATE TRIGGER . . . . . Y (Y,N,D)  REBIND PLAN/PACKAGE  . . . . Y (Y,N,D)
  CREATE MASK  . . . . . . Y (Y,N)    ALTER TABLE ACTIVATE CONTROL Y (Y,N)
  CREATE PERMISSION  . . . Y (Y,N)
  CREATE STORAGE GROUP . . Y (Y,N)    GRANT use OF STORAGE GROUP . Y (Y,N,A,R)

 New names/values for generated SQL: (leave blank to use current values)
   Object schema  . . . . .           > Run SQLID  . . . . . . . .
   Object grantor . . . . .           >
   Alloc TS size as . . . . DEFINED     (DEFINED, USED, or ALLOC)
   Database name  . . . . .
   Storage group for TS . .           > Storage group for IX . . .            >
   Target DB2 version . . .             (Current DB2 version: 1115)
   Use Masking  . . . . . . NO          (Yes/No)
   Use Exclude Spec . . . . NO          (Yes/No)
   Target cat qualifier . .           > (Default is SYSIBM)
   Generate catalog stats . NO          (Yes,No,Only)
     Statistics tables  . . ALL         (All or Select. Default is All)
   Include DB2 pending chgs NO          (Yes,No,Alter,Only)
   PBG NUMPARTS value . . . EXISTING    (Defined, Existing)
   PBG LOB objects  . . . . COMPUTED    (Computed, Implicit)
   Generate index cleanup .             (Yes,No,Only)

 SQL output data set and execution mode:
   Add to a WSL . . . . . . NO          (Yes/No)
   Data set name  . . . . .
     Data set disposition . OLD         (OLD, SHR, or MOD)
   Execution mode . . . . . BATCH       (BATCH or TSO)
   Commit statements per  .             (Db, tS, Tb, All, None. Default is All)
   DB2 defaults handling  .             (Keep, or Remove. Default is Keep)
   Prompt to run SQL  . . . NO          (Yes/No. For TSO mode and no WSL)
   Include SQL comments . . NO          (Yes/No. For BATCH mode and no WSL)

 DB2 Command output data set:
   Data set name  . . . . .
     Data set disposition . OLD         (OLD, SHR, or MOD)

 BP - Change batch job parameters
  G - Change additional parameters
```

Figure 60. Generate SQL from DB2 Catalog panel (ADB2GEN)

Press Enter to display the reverse engineering output. The following figure shows part of the result of reverse engineering this database.

```
-----------------------------------------------------------------------
--                                                                   --
-- Database 2 Administration Tool (DB2 Admin) , program 5655-DAT (C) --
--                                                                   --
-- ADB2GEN  -  Extract object definitions from the DB2 Catalog tables --
--                                                                   --
-- Input prepared on : DB2X (810)     Extract time : 2013-16-04 01:01 --
--                                                                   --
-- Catalog values overridden : none                                  --
--                                                                   --
-- Generate : SG=Y DB=Y TS=Y TB=Y VW=Y IX=Y SY=Y AL=Y LB=Y CM=Y FK=Y  --
--            TG=Y UT=N UF=N SP=N                                     --
-- Grants   : SG=Y DB=Y TS=Y TB=Y VW=Y SC=N UT=N UF=N SP=N            --
--                                                                   --
-----------------------------------------------------------------------
-----------------------------------------------------------------------
--                                                                   --
-- ADB2GEN: Generate DDL for Database DSN8D81A                        --
--                                                                   --
-----------------------------------------------------------------------
--
--
-----------------------------------------------------------------------
-- Database=DSN8D81A    Stogroup=DSN8G810
-----------------------------------------------------------------------
--
  SET CURRENT SQLID='DSCGDB2';
--
  CREATE DATABASE DSN8D81A
    BUFFERPOOL BP0
    INDEXBP    BP2
    CCSID      EBCDIC
    STOGROUP DSN8G810 ;
--
  GRANT DBADM
    ON DATABASE DSN8D81A TO PUBLIC;
--
  COMMIT;
--
```

Figure 61. Reverse engineering output (1 of 2)

```
-------------------------------------------------------------------------
-- Database=DSN8D81A   Stogroup=DSN8G810
-- Tablespace=DSN8D81A.DSN8S81D
-------------------------------------------------------------------------
--
  CREATE TABLESPACE DSN8S81D
    IN DSN8D81A
    USING STOGROUP DSN8G810
    PRIQTY 32 SECQTY 20
    ERASE  NO
    FREEPAGE 0 PCTFREE 5
    GBPCACHE CHANGED
    TRACKMOD YES
    BUFFERPOOL BP0
    LOCKSIZE PAGE
    LOCKMAX SYSTEM
    CLOSE NO
    COMPRESS NO
    CCSID       EBCDIC
    MAXROWS 255;
--
  GRANT USE OF TABLESPACE DSN8D81A.DSN8S81D TO PUBLIC;
--
  COMMIT;
--
-------------------------------------------------------------------------
--    Table=DSN8810.DEPT              In DSN8D81A.DSN8S81D
-------------------------------------------------------------------------
--
  SET CURRENT SQLID='DSN8810';
--
  CREATE TABLE DSN8810.DEPT
     (DEPTNO             CHAR(3) FOR SBCS DATA NOT NULL ,
      DEPTNAME           VARCHAR(36) FOR SBCS DATA NOT NULL ,
      MGRNO              CHAR(6) FOR SBCS DATA WITH DEFAULT NULL ,
      ADMRDEPT           CHAR(3) FOR SBCS DATA NOT NULL ,
```

Figure 62. Reverse engineering output (2 of 2)

If you specify Y for REBIND PLAN/PACKAGE on the Generate SQL from DB2 Catalog Panel, shown in Figure 60 on page 155, the following output is also displayed.

```
Command ===>                                           Scroll ===> PAGE
****** ***************************** Top of Data ******************************
000001   REBIND PACKAGE(DSN8ES81.DSN8ES1)
****** **************************** Bottom of Data ****************************
```

Figure 63. Reverse engineering rebind output

Chapter 7. Setting DB2 Admin parameters

You can change the default settings and appearance of DB2 Admin panels.

Topics:
- "Using the DB2 Admin Options panel"
- Changing ISPF settings
- "Changing colors and highlights"
- "Changing DB2 Admin defaults" on page 160
- "Changing/Allocating print data sets" on page 166
- "Changing DB2 Admin prompt options" on page 171
- "Changing migrate settings" on page 165
- Changing the SQL ID

Using the DB2 Admin Options panel

Use the DB2 Admin Options panel to select the DB2 Admin parameters that you want to change.

Select option P on the Administration Menu panel to display the DB2 Admin Options panel, as shown in the following figure. Alternatively, you can invoke the panel by issuing the OPTIONS primary command on any DB2 Admin panel.

Use this panel to select the DB2 Admin parameters that you want to change. To select a category of parameters, enter the corresponding option in the **Option** field and press Enter.

```
DB2 Admin                        DB2 Admin Options                    11:10
 Option ===>

   1 - Colors and highlights                         DB2 System: DB2X
   2 - DB2 Admin defaults                            DB2 SQL ID: ISTJE
   A - Alter options
  BP - Batch parameters
  CH - Options for change functions
   D - Display options
   I - Installation default parameters
   G - Generate parameters
   M - Migrate options
   P - Print data set options
  PR - Prompt options
  SV - Manage session scope variables
```

Figure 64. DB2 Admin Options panel (ADB2P)

Changing colors and highlights

Use the Change Colors and HighlightColors and Highlight panel to change the colors or highlighting scheme (or designations) technique on DB2 Admin panels.

Select option 1 on the DB2 Admin Options panel to display the Change Colors and HighlightColors and Highlight panel, as shown in the following figure.

Use this panel to change the colors or highlighting scheme (or designations) technique on DB2 Admin panels.

If you leave an input field on the panel blank, the default value is used. Specify RESET on the command line to choose default values for all sections of the panel.

```
DB2 Admin                   Colors and Highlights                    15:46
Command  ===>

 DB2 Admin panels consist of standard sections, as listed below.
 Select colors and highlights to use for each section.

 Valid Colors     :  yellow red blue green white pink and turq
 Valid Highlights :  blink reverse uscore or blank (default)

                        Color:              Highlight:
 Headings:              YELLOW
 Text:                  BLUE
 Highlighted text:      TURQ
 Messages:              RED
 Function:              WHITE
 Input areas:           GREEN
 Output areas:          TURQ
 Scrollable fields:     BLUE
 Scrollable columns:    BLUE

 Press ENTER to activate changes or PF3 to cancel changes.
```

Figure 65. Colors and Highlight panel (ADB2P1)

The fields on this panel are:

Headings
: First line of the panel (the default setting is yellow)

Text
: Instructions or descriptions on the panel (default is blue)

Highlighted text
: Emphasized text (the default setting is turquoise)

Messages
: Message area, third line on the panel when a message is returned (the default setting is red)

Function
: Command line and/or option chosen (the default setting is white)

Input areas
: Area in which you enter your input (the default setting is green)

Output areas
: Area in which output is returned to you (the default setting is turquoise)

Scrollable fields
: Fields that you can scroll for more information (the default setting is blue)

Scrollable columns
: Columns that you can scroll for more information (the default setting is blue)

Changing DB2 Admin defaults

Use the Change DB2 Admin Defaults panel to change various parameters that affect the execution of DB2 Admin.

Select option 2 on the DB2 Admin Options panel to display the DB2 Admin Defaults panel, as shown in the following figure.

Use this panel to change various parameters that affect the execution of DB2
Admin.

```
ADB2P2 in                    DB2 Admin Defaults                    12:16
Option ===>

                                             DB2 System: DSNA
                                                        More:     +

Max No of Rows to Fetch  . . . . . . 1000   (0-327670, 0=unlimited, def. 1000)

Max Chars in an SQL Stmt . . . . . . 32765  (4000-32765, default is 32765)

Pgm Action when SQL error:
  First do a . . . . . . . . . . . R        (C-Commit, R-Rollback)
  Display error panel  . . . . . . . YES    (Yes/No)
  Continue executing SQL . . . . . . NO     (Yes/No)

Auto Refresh After Update  . . . . . YES    (Yes/No, default is YES)

Display SQL cost estimate  . . . . . NO     (Yes/No, default is YES)

Browse DB2 Command Output  . . . . . YES    (Yes/No)

Max Chars in an ISPF Stmt  . . . . . 2000   (500-32765, default is 2000)

Max Chars in an Admin Cmd  . . . . . 32765  (500-32765, default is 32765)

Report Drop Impacts  . . . . . . . . YES    (Yes/No)

Report Revoke Impacts  . . . . . . . YES    (Yes/No)

Reset to Def. at Startup . . . . . . NO     (Yes/No)

Action when no rows found  . . . . . M      (M - Message (default), P - Panel)

Default local CCSID  . . . . . . . . 00000  (Optional, numeric)

Verify CCSID . . . . . . . . . . . . YES    (Yes/No, default is YES)

Capitalize object names  . . . . . . YES    (Yes/No, default is YES)

Capitalize data  . . . . . . . . . . YES    (Yes/No, default is YES)

Use trusted context in batch . . . . NO     (Yes/No, default is NO)

Gen. utilities for restricted  . . . YES    (Yes/No, default is YES)

Line command field behavior  . . . .        (*CMD, *, Clear , default is *CMD)

Display result of explain  . . . . . NO     (Yes/No, default is NO)

CAT command character  . . . . . . . ?      (default is question mark)

Query type for views . . . . . . . . E      (E-Enhanced or S-Singular)

Prefix for LOB files . . . . . . . .         (Prefix/blank, def. is blank)

Limit for LOB data . . . . . . . . . 16     (Number of MB, 1 - 256, def. is 16)

Query Java SP package  . . . . . . . D      (D-Default or E-Enhanced)

Get DB2 ZPARM  . . . . . . . . . . . YES    (Yes/No)

Format type for SQL stmts  . . . . . E      (E-Enhanced or S-Simple)

Run Accelerator functions in batch . NO     (Yes/No, default is NO)
```

Figure 66. Change DB2 Admin Defaults panel (ADB2P2)

The fields on this panel are fully described in the help panel. Some of the fields are
as follows:

Max No of Rows to Fetch
Enter the maximum number of rows to fetch for each SQL SELECT statement. A high value for this field can result in long response times for "wild" queries.

Max Chars in an SQL Stmt
Enter the maximum length of the buffer for SQL and ISPF statements. DB2 Admin allocates this number of bytes when displaying a new panel. A high value for this field can cause slow TSO performance on a storage constrained system.

Pgm Action when SQL error
Specify the action that DB2 Admin takes when an SQL error occurs. The choices are:
- COMMIT or ROLLBACK the changes
- Display the SQL error panel with the SQL error message and SQLCA (YES or NO)
- Continue processing by executing the next SQL statement (YES or NO)

Auto Refresh After Update
Indicate whether table display panels are to be refreshed after SQL updates (YES or NO). If YES, DB2 Admin refreshes the panels when they are redisplayed. For performance reasons, the refresh is limited to panels where the elapsed time to fetch the rows to be displayed is less than 10 seconds. A value of NO for this field can result in you viewing and acting on old data when you press END.

Display SQL cost estimate
Specify whether you want DB2 Admin to display an estimated cost for an SQL SELECT statement. The estimate is displayed as an ISPF message. If the estimated cost is larger than the maximum value of an integer, the estimated cost is displayed as "*.***.***.***".

Browse DB2 Command Output
Indicate whether DB2 Admin should invoke ISPF browse (YES) or let the output default to TSO line mode (NO).

Max Chars in an ISPF Stmt
Enter the maximum length of the buffer for ISPF statements. A high value for this field can cause slow TSO performance on a storage constrained system.

Max Chars in an Admin Cmd
Enter the maximum length of the buffer for DB2 Admin commands. A high value for this field can cause slow TSO performance on a storage constrained system.

Report Drop Impacts
Enter the default value to be displayed in the **Report Drop Impacts** field when dropping an object.

Report Revoke Impacts
Enter the default value to be displayed in the **Report Revoke Impacts** field when revoking authorities.

Reset to Def. at Startup
Indicate whether DB2 Admin should restore the following parameters to their default values at the next startup:
- MAX NO OF ROWS TO FETCH
- MAX CHARS IN AN SQL STATEMENT
- AUTO REFRESH AFTER UPDATE

- MAX CHARS IN AN ISPF STMT
- MAX CHARS IN AN ADMIN CMD

When set to NO, DB2 Admin attempts to restore the CURRENT SQLID.

Action when no rows found
Indicate whether DB2 Admin displays a pop-up panel (P) or just a message (M) when no rows are found.

Default local CCSID
If the ISPF system or terminal emulator are set up such that no CCSID is available in ZTERMCID, specify a default to enable the SQ line command for packages, plans and triggers that are created in DB2 Version 8 or higher.

Verify CCSID
Indicate whether DB2 Admin verifies that the coded character set identifier (CCSID) for the TSO terminal and the CCSID for the plan under which DB2 Admin is running match each other. When you start DB2 Admin and verification is active, a pop-up panel is displayed to provide a warning if the CCSIDs do not match. (The pop-up panel is also displayed when you start DB2 Object Comparison Tool and the CCSIDs of the TSO terminal and the plan under which DB2 Object Comparison is running do not match each other.) A discrepancy in the CCSIDS can lead to unexpected data conversion, affecting any characters that do not map to the same code point in the two CCSIDs.

Capitalize object names
Indicates whether DB2 Admin translates the lowercase characters that you use in object names, qualifiers, and authorization identifiers in the following fields on the System Catalog panel (ADB21) to uppercase characters:

- Name
- Owner
- In D/L/H (databases, collections, and schema)
- Grantor
- Grantee

When the value of the parameter is NO, lowercase characters that are specified in these fields are not translated to uppercase characters unless the object being displayed is restricted to having a name with uppercase characters only according to the rules of DB2. For example, database names, table space names, plan names, and package names (except for trigger package names) must have names in uppercase characters and, therefore, will always be translated to uppercase characters.

When the value of this parameter is NO, DB2 Admin also supports the use of lowercase characters in the qualifier and name of the object when you use DB2 Admin panels to:
- Create or drop an index.
- Create or drop an view.
- Drop a table.

pan

Capitalize data
Indicates whether DB2 Admin translates the lowercase characters that you enter as data to uppercase characters.

Use trusted context in batch
Indicates whether the ASUSER parameter that is used in the online function should also be used in batch.

Gen. utilities for restricted
Specifies whether DB2 Admin should prompt for additional utilities when DB2 places an object in an restrictive state and returns SQLCODE +610.

Display result of explain
Displays the PLAN_TABLE rows if EXPLAIN MODE is on and YES is entered for the `Display result of explain` field.

CAT command character
Specifies a character that can be used as a shortcut for the CAT command. The character cannot be alphanumeric or the current value of the ISPF command delimiter. Other character restrictions are detailed in the help information.

Prefix for LOB files
High level qualifier(s) for LOB files. Specifies the prefix for temporary LOB files. The default is blank. If the prefix contains a period the TSO prefix is not appended to the file name following the specified prefix.

Query Java SP package
Specifies the algorithm to use for locating the packages of a Java stored procedure, when the K line command is issued on the Stored Procedures panel (ADB21O).

D   Packages are located by using the COLLID value and EXTERNAL NAME value of the Java stored procedure, which are stored in the DB2 catalog tables. D is the default.

E   Packages are located by using the default algorithm with the following enhancement:

If no packages are found, the DB2 Admin Tool attempts to locate packages by using the COLLID value and CLASS value of the stored procedure.

If the CLASS is embedded in the REMARKS column of a package, then the DB2 Admin Tool associates the package with the stored procedure, provided that one of the following conditions is true: 1) The COLLID value of the package is equal to the COLLID value of the stored procedure. 2) The COLLID value of the package is NULLID if the COLLID value of the stored procedure is blank.

Get DB2 ZPARM
Specifies whether to call DSNWZP in a process. The default value is YES.

DSNWZP requires DB2 Monitor privileges. When DB2 is on V10 NFM or a higher version, DB2 Admin does not call DSNWZP during the bind or rebind process.

Format type for SQL stmts
Specifies the format for displaying SQL statements.

S   Displays SQL statements in simple format, with chunks of 72 byte text on one line and host variable information on additional lines.

E   Displays SQL statements with complex nested subqueries in enhanced format. Only DECLARE CURSOR and SELECT statements can be displayed in the enhanced format. All other statement types are displayed in simple format.

Run Accelerator functions in batch
Specifies whether accelerator functions are run in batch (YES) or in TSO (NO).

Changing alter options

Use the Alter Options panel to change settings for the ALTER command.

Select option A on the DB2 Admin Options panel to display the Alter Optionspanel.

Changing batch parameters

Use the Batch Job Utility Parameters panel to change batch job settings.

Select option BP on the DB2 Admin Options panel to display the Changing batch parameterspanel.

Options for change functions

Use the Options for change functions panel to change settings that are common to change functions.

Select option CO on the DB2 Admin Options panel to display the Change options common to change functions panel.

Display options

Use the Display panel to customize the display for supported table display panels.

Select option D on the DB2 Admin Options panel to display the Display options panel.

Changing installation default parameters

Use the Changing installation default parameters panel to set global values for the **PARALLEL** parameter.

Select option I on the DB2 Admin Options panel to display the Installation Defaults panel.

Generating parameters

Use the Generating parameters panel to manage the Generate function.

Select option 1 on the DB2 Admin Options panel to display the Generate parameters panel.

Changing migrate settings

Use the migrate function to change the parameter that controls whether space information is gathered and displayed in the Migrate Table Spaces panel (ADB28S).

About this task

To change the parameter that controls whether space information is displayed:

Procedure

1. Select option M on the Change DB2 Admin Settings panel. The Change Migrate Settings panel is displayed.
2. Specify YES or NO in the **Show space information on panels** field.

Changing/Allocating print data sets

Use the Change/Allocate Print Data Set panel to allocate a print data set for the DB2 Admin print function.

Select option P on the DB2 Admin Options panel to display the Print Data Set Options panel, as shown in the following figure.

Use this panel to allocate a print data set for the DB2 Admin print function.

```
 DB2 Admin                  Print Data Set Options                00:27
 Option ===>

 Enter data set name and disposition:
   Data set name . .
   Disposition   . . .              (NEW,OLD,MOD,FREE)

 Enter attributes for a NEW data set:
   Lrecl  . . . . . .            (8-32760)
   Block size . . . .            (0-32760)
   Format . . . . . .            (Fixed or Variable)
   Space units  . . .            (Tracks, Cylinders or Blocks)
   Primary space  . .            (Default 1)
   Sec. space . . . .            (Default 1)
   Unit type  . . . .            (Default SYSDA)
```

Figure 67. Print Data Set Options panel (ADB2PP)

The fields on this panel are:

Enter data set name and disposition
Enter the name and allocation mode of the print data set, as described below.

   Data set name
   Enter the name of the data set that DB2 Admin should use for printing.

   Disposition
   Enter the allocation mode of the data set, which must be one of the following values:

      NEW
      Allocate a new data set.

      OLD
      Use an existing data set.

      MOD
      Append output to an existing data set.

      FREE
      Deallocate print data set.

For a NEW data set enter:
For a new data set, the following parameters are required:

   Lrecl
   Specify the logical record length.

   Block size
   Specify the block size.

   Format
   Specify the data set format, which can be either F (for fixed) or V (for variable) length records.

Space units
Specify the units in which space is to be allocated (tracks, cylinders, or blocks).

Primary space
Specify the primary space allocation, specified in preceding units.

Sec. space
Specify the secondary space allocation, specified in preceding units.

Unit type
Specify the type of UNIT for allocation.

Example: Printing ISPF table content to a data set

The following example demonstrates how to use the DB2 Admin print function to capture the contents of an ISPF table to a data set.

Step 1: Create the file that you want to send content to

Determine the format that you want for your data set based on the data that you want to store. In this example, the data set name is NEWONE.SAMPLE.PRINT. NEWONE is the qualifier.

```
ISRUAIES DSLIST                      Data Set Information
Command ===>

Data Set Name . . . . : NEWONE.SAMPLE.PRINT

General Data                          Current Allocation
Management class . . : PRIMARY        Allocated cylinders : 1
Storage class  . . . : NORMALG        Allocated extents . : 1
 Volume serial . . . : SM4225
 Device type . . . . : 3390
Data class . . . . . : **None**
 Organization  . . . : PS             Current Utilization
 Record format . . . : FB              Used cylinders  . . : 0
 Record length . . . : 133             Used extents  . . . : 0
 Block size  . . . . : 27930
 1st extent cylinders: 1
 Secondary cylinders : 1              Dates
 Data set name type  :                 Creation date . . . : 2013/08/27
                                       Referenced date . . : ***None***
                                       Expiration date . . : ***None***
 SMS Compressible  . : NO
```

The following fields control the format of the data set:

Organization
Physically sequential (PS).

Record format
Fixed block (FB).

Record length
LRECL 133.

Block size
BZSIZE 27930.

Step 2: Allocate the data set in the PRINT data definition (DD)

You can allocate the data set in the PRINT DD or PRTTAB DD either through a logon procedure or the TSO ALLOC command. For example, you can run the following command: TSO ALLOC F(PRINT) DSN('NEWONE.SAMPLE.PRINT') OLD

The print data set can also be allocated within DB2 Admin by using the option P.P to access the following panel:

```
DB2 Admin -------------- Change/Allocate Print Data Set ----------------- 07:14
Option ===>

Enter data set name and disposition:
  Data set name ===> 'NEWONE.SAMPLE.PRINT'
  Disposition   ===> NEW        (NEW,OLD,MOD,FREE)

For a NEW data set enter:
  Lrecl         ===> 133        (8-32760)
  Block size    ===> 27930      (0-32760)
  Format        ===> F          (Fixed or Variable)
  Space units   ===> T          (Tracks, Cylinders or Blocks)
  Primary space ===>            (Default 1)
  Sec. space    ===>            (Default 1)
  Unit type     ===>            (Default SYSDA)
```

In the panel, you allocate the data set to DD-name (file) PRINT in preparation for using the print command: PRT TABLE ON FILE PRINT.

Step 3: View what you want to print

In this example, the content that is to be printed is a package list. In DB2 Admin, you navigate to the object that you want to print.

```
ADB21P in ---------------- DSNA Application Plans ------------ Row 1 to 1 of 1
Command ===>                                            Scroll ===> CSR

Commands: BIND  REBIND  FREE  GRANT
Line commands:
DP - Depend  A - Auth  T - Tables  V - Views  X - Indexes  S - Table spaces
Y - Synonyms  M - DBRMs  RB - Rebind  F - Free  B - Bind  GR - Grant
PL - Package list  LP - List PLAN_TABLE  I - Interpret  ENDI - Enab/disab con
K - Local packages  SQ - SQL  D - Databases  RO - Role
                      Bind    Bind   V I V O Bound   Quali-    Pack A R E D
Select Name    Owner   Date   Time   D S A P By       fier     Lists Q L X R
     *       *        *      *      * * * * *        *          * * * * *
------ -------- -------- ------ ------ - - - - -------- -------- ------ - - - -
PL     ADBDEV   K351156  130826 163416 B S Y Y J148286  DB2ADM     13 U C N
***************************** END OF DB2 DATA *****************************
```

```
ADB21PL n ----------------- DSNA Package List ------------- Row 1 to 13 of 13
Command ===>                                           Scroll ===> PAGE

Line commands:  K - Local packages  I - Interpretation

S PL Name  Seq No Location          Collection          Name     Timestamp
  *          * *                    *                   *        *
- -------- ------ ---------------- ------------------ -------- ----------------
  ADBDEV      1 *                   VB1DEV0             *        2013-08-26-16.34
  ADBDEV      2 *                   VB1APAR             *        2013-08-26-16.34
  ADBDEV      3 *                   ADBB1PAR            *        2013-08-26-16.34
  ADBDEV      4 *                   ADBB1MPE            *        2013-08-26-16.34
  ADBDEV      5 *                   VA2APAR             *        2013-08-26-16.34
  ADBDEV      6 *                   ADBA2PAR            *        2013-08-26-16.34
  ADBDEV      7 *                   ADBA2MPE            *        2013-08-26-16.34
  ADBDEV      8 *                   V10APAR             *        2013-08-26-16.34
  ADBDEV      9 *                   ADB10PAR            *        2013-08-26-16.34
  ADBDEV     10 *                   ADB10MPE            *        2013-08-26-16.34
  ADBDEV     11 *                   V72APAR             *        2013-08-26-16.34
  ADBDEV     12 *                   ADB72PAR            *        2013-08-26-16.34
  ADBDEV     13 *                   ADB72MPE            *        2013-08-26-16.34
****************************** END OF DB2 DATA ********************************
```

Step 4: Issue the command PRINT TABLE ON FILE

In the panel that contains the object that you want to print, you issue the print
command: >PRINT TABLE ON FILE PRTTAB. The TSO command prefix (>) is used to
prevent the TSO PRINT command from running in conflict with the PRINT TABLE
ON FILE command.

```
ADB21PL n ----------------- DSNA Package List ------------- Row 1 to 13 of 13
Command ===> >PRINT TABLE ON FILE PRTTAB              Scroll ===> PAGE

Line commands:  K - Local packages  I - Interpretation

S PL Name  Seq No Location          Collection          Name     Timestamp
  *          * *                    *                   *        *
- -------- ------ ---------------- ------------------ -------- ----------------
  ADBDEV      1 *                   VB1DEV0             *        2013-08-26-16.34
  ADBDEV      2 *                   VB1APAR             *        2013-08-26-16.34
  ADBDEV      3 *                   ADBB1PAR            *        2013-08-26-16.34
  ADBDEV      4 *                   ADBB1MPE            *        2013-08-26-16.34
  ADBDEV      5 *                   VA2APAR             *        2013-08-26-16.34
  ADBDEV      6 *                   ADBA2PAR            *        2013-08-26-16.34
  ADBDEV      7 *                   ADBA2MPE            *        2013-08-26-16.34
  ADBDEV      8 *                   V10APAR             *        2013-08-26-16.34
  ADBDEV      9 *                   ADB10PAR            *        2013-08-26-16.34
  ADBDEV     10 *                   ADB10MPE            *        2013-08-26-16.34
  ADBDEV     11 *                   V72APAR             *        2013-08-26-16.34
  ADBDEV     12 *                   ADB72PAR            *        2013-08-26-16.34
  ADBDEV     13 *                   ADB72MPE            *        2013-08-26-16.34
****************************** END OF DB2 DATA ********************************
```

Step 5: Select the content that you want to print and exit

In the Print Layout (ADB2DPRT) panel, you can select the columns of data that
you want to print:

```
ADB2DPRT  ------------------ DSNA Print Layout --------------- Row 1 to 7 of 7
Command ===>                                            Scroll ===> PAGE

Current print columns:


Select Column Name         Col No Col Type Length Scale
       *                      * *  --------    *     *
------ -----------------    ------ -------- ------ ------
S      PLANNAME                1 VARCHAR      24     0
S      SEQNO                   2 SMALLINT      2     0
S      LOCATION                3 VARCHAR     128     0
       COLLID                  4 VARCHAR     128     0
       NAME                    5 VARCHAR     128     0
       TIMESTAMP               6 TIMESTMP     26     0
       IBMREQD                 7 CHAR          1     0
***************************** END OF DB2 DATA ******************************
```

```
ADB2DPRT  ------------------ DSNA Print Layout --------------- Row 1 to 7 of 7
Command ===>                                            Scroll ===> PAGE

Current print columns:
(PLANNAME SEQNO LOCATION)

Select Column Name         Col No Col Type Length Scale
       *                      * *  --------    *     *
------ -----------------    ------ -------- ------ ------
*      PLANNAME                1 VARCHAR      24     0
*      SEQNO                   2 SMALLINT      2     0
*      LOCATION                3 VARCHAR     128     0
       COLLID                  4 VARCHAR     128     0
       NAME                    5 VARCHAR     128     0
       TIMESTAMP               6 TIMESTMP     26     0
       IBMREQD                 7 CHAR          1     0
***************************** END OF DB2 DATA ******************************
```

Result: View the data set

In the standard Browse data panel (ISRBROBA) in z/OS ISPF, you can view the data set.

```
ISRBROBA  NEWONE.SAMPLE.PRINT                    Line 00000000 Col 001 080
Command ===>                                            Scroll ===> CSR
***************************** Top of Data *********************************
PLANNAME  SEQNO LOCATION
--------  ------ --------
ADBDEV       1 *
ADBDEV       2 *
ADBDEV       3 *
ADBDEV       4 *
ADBDEV       5 *
ADBDEV       6 *
ADBDEV       7 *
ADBDEV       8 *
ADBDEV       9 *
ADBDEV      10 *
ADBDEV      11 *
ADBDEV      12 *
ADBDEV      13 *
***************************** Bottom of Data ******************************
```

Changing DB2 Admin prompt options

Use the Prompt Options panel to change DB2 Admin prompt options.

Select option PR on the Change DB2 Admin Settings panel to display the Prompt Options panel, as shown in the following figure. Use the Prompt Options panel to change DB2 Admin prompt options. By turning on the prompt option, you are prompted before certain SQL statements are run. Specify YES to activate prompting on the options listed in the following figure.

```
DB2 Admin ---------------------- Prompt Options ---------------------- 01:52
Option ===>

Change one or more options below. Prompt before executing:

  Definition SQL (CREATE, DROP, ALTER, RENAME,..)  ===> NO  (Yes/No)

  Authorization SQL (GRANT and REVOKE)             ===> YES (Yes/No)

  Update SQL (INSERT, UPDATE, DELETE)              ===> NO  (Yes/No)

  DSN commands (BIND, REBIND and FREE)             ===> NO  (Yes/No)

  DB2 commands (START, STOP, ALTER, SET)           ===> NO  (Yes/No)
```

Figure 68. Prompt Options panel (ADB2PRMT)

The fields on this panel are:

Definition SQL
> Any SQL statement that changes the definition of an object, such as CREATE, ALTER, DROP, and RENAME

Authorization SQL
> GRANT and REVOKE SQL statements

Update SQL
> INSERT, UPDATE, and DELETE statements

DSN commands
> A DSN command statement, such as BIND, REBIND, or FREE

DB2 commands
> A DB2 command that changes the state of an object or the system

When any of the prompt options are used, the Statement Execution Prompt panel is displayed, as shown in the following figure. For example, in the previous figure, prompting before running authorization statements is requested. The following figure shows the prompt panel that is displayed before running a request to grant load access to database TESTDB01.

```
DB2 Admin -------------- DB2X Statement Execution Prompt -------------- 11:46
Option ===>

DB2 Admin is about to execute the statement below. You have asked to be
prompted before DB2 Admin executes this type of statement. What do you want to
do now:
    1 - Execute the statement
    2 - Edit the statement
    3 - Create a batch job with the statement
    4 - Add the statement to the work statement list
 CAN - Cancel
 Work statement list dsn  ===>
 Work statement list name ===>              Action ===> A (Append or Replace)
                                                            More:      +
Statement that is about to be executed (first 28 lines):
GRANT LOAD
  ON DATABASE TESTDB01
  TO ISTJE
```

Figure 69. Statement Execution Prompt panel (ADB2PSTM) – granting authorizations

When more than one SQL statement is to be run, the message (add an A for all
stmts. For example 1A – Execute all stmts) is issued on the Statement
Execution Prompt panel, and the following additional options are available:

1A Runs all statements.

3A Runs all statements in batch mode.

4A Adds all statements to a work statement list. If the action is append, the
        statements are added to the end of the work statement list. If the action is
        replace, the work statement list is erased and then the statements are added.

Managing session scope for global variables

Use the Session Scope Variables panel to create and manage global variables that
you want to apply only to the current session.

Session scope variables override the default values of global variables.

The following conditions apply to session scope variables:
- They can be DB2 built-in global variables or user-defined global variables.
- They are active only for the current session and only when explicitly set.
- Their values may be used in the normal processing of other SQL statements such
  as DELETE, INSERT, SELECT, or UPDATE.

Select option SV on the DB2 Admin Options panel to display the Session Scope
Variables panel, as shown in the following figure.

```
  ╭─────────────────────────────────────────────────────────────────────────────────╮
  │ DB2 Admin                 DB2X Session Scope Variables                      15:40 │
  │ Option ===>                                                                       │
  │                                                                                   │
  │  Use these variables to override global variables for    DB2 System: DB2X        │
  │  this session.                                           DB2 SQL ID: SYSADM       │
  │                                                                                   │
  │    SV - Manage session scope variables                                            │
  │                                                                                   │
  │   CSV - Create session scope variable control table                               │
  │   USV - Upgrade session scope variable control table                              │
  │                                                                                   │
  │   Session scope variable control table:                                           │
  │     Table schema  . . SYSADM     >                                                │
  │     Table name  . . . GLOBALVAR       >                                           │
  ╰─────────────────────────────────────────────────────────────────────────────────╯
```

Figure 70. Session Scope Variables panel (ADBPPSV)

Option SV displays the Session Scope Variables in schema.table panel, as shown in
the following figure.

```
  ╭─────────────────────────────────────────────────────────────────────────────────╮
  │ DB2 Admin     DB2X Session Scope Variables in SYSADM.GLOBA  >   Row 1 to 2 of 2   │
  │ Command ===>                                                   Scroll ===> CSR    │
  │                                                                                   │
  │ Line commands:  U - Update  SET - Set variable  DEL - Delete  INS - Insert        │
  │                 GV - Global variables                                             │
  │                                                                                   │
  │ Select Schema    Name            Expression                                       │
  │        *         *               *                                                │
  │ ------ -------> --------------> --------------------------------------------       │
  │        SYSIBMAD GET_ARCHIVE     Y                                                  │
  │        SYSIBMAD MOVE_TO_ARCHIVE Y                                                  │
  │ ****************************** END OF DB2 DATA********************************     │
  ╰─────────────────────────────────────────────────────────────────────────────────╯
```

Figure 71. Session Scope Variables in schema.table panel (ADBPPSV1)

Use line commands to manage the session variables.

Chapter 8. Querying the system catalog

You can use the main System catalog panel to query the DB2 system catalog.

You can perform the following tasks:
- Display any object in the DB2 catalog
- Display related DB2 objects using DB2 Admin line commands
- Interpret catalog information
- Show the authorizations for DB2 objects
- Display the static SQL statements from application plans and packages
- Display the DDL for existing views
- Generate JCL (job control language) for the DB2 utilities and then run them online
- Execute dynamic SQL statements
- Issue DB2 commands for databases and database objects
- Display database structures
- Reverse engineer DB2 objects
- Generate reports about the DB2 objects that are saved in a printable format

For more information about using the System catalog panel, see "The System Catalog panel" on page 969.

Topics:
- "Using a copy of the DB2 catalog"
- "Selecting a copy of the DB2 catalog"
- "Creating reports from the DB2 catalog" on page 176
- "Redefined columns in the DB2 catalog" on page 179
- "DB2 Admin restrictions on DB2 object names" on page 180

Using a copy of the DB2 catalog

If your subsystem supports using multiple copies of the DB2 catalog, you can use the System Catalog panel **Switch Catalog Copy** field at the bottom of the panel to switch between copies of the catalog.

Valid values include:

N   No change. Continue using the same catalog.

S   Use the system catalog.

C   Use a copy of the DB2 system catalog. When you choose this option, the Select Copy of DB2 Catalog panel is displayed. On this panel, select a catalog. The suffix *xx* in CC*xx* is the plan name suffix assigned to the copy. In the heading of all subsequent system catalog panels, CC*xx* is displayed instead of the DB2 subsystem name.

Selecting a copy of the DB2 catalog

Use the System Catalog panel to select a copy of the DB2 catalog.

The Select Copy of DB2 Catalog panel is displayed, as shown in the following figure, when you enter C on the Switch Catalog Copy line on the System Catalog panel. The panel shows a list of copies of the DB2 system catalog; select one of

them by entering an S in front of the appropriate catalog.

```
DB2 Admin -------------- DB2X Select Copy of DB2 Catalog ----------------------
Command ===>                                               Scroll ===> PAGE

DB2 Catalog Copy Version Selection:                    DB2 System: DB2X
                                                       DB2 SQL ID: ISTJE
 S - Select an entry
                                   Copy     Planname
Select Timestamp                   Owner    Suffix   Type Location
       *                           *        *        *    *
------ -------------------------- -------- -------- ---- ----------------
       2004-01-09-18.17.27.341202 COPY02   02        C
       2004-01-20-14.49.07.032221 COPY01   01        C
       ?                          ALIES2   A2        A    SYSTEM4A_DB2X
       ?                          ALIES6   A6        A    SYSTEM4A_DB2X1
       ?                          COPY03   03        C
***************************** END OF DB2 DATA *************************
```

Figure 72. Select Copy of DB2 Catalog panel (ADB2CCS)

The panel includes the following columns:

Select Input field in which you enter the S line command to select a catalog.

Timestamp
Time when the copy of the catalog was last refreshed.

Copy Owner
The user ID that owns the catalog copy.

Planname Suffix
Suffix that identifies the catalog. When a copy of the DB2 catalog is used, this suffix is used on the header of the system catalog panels instead of the DB2 subsystem identifier.

Type Type of catalog. The catalog can be one of the following types:

> **A** Alias of a (distributed) DB2 system catalog.
>
> **C** Copy of the local DB2 system catalog.

Location
Name or location of the remote DB2 subsystem.

Creating reports from the DB2 catalog

You can create reports about the objects in the DB2 catalog that can be saved and printed.

Overview of reports

You can use the REP command to generate reports that can be saved and printed.

When you use the system catalog panels to display information about the objects in the DB2 catalog, you can use the REP command to generate reports with information (that is similar to the displayed information) that can be saved and printed.

Reviewing printed reports can be faster than stepping through the information online. Saving reports about your databases at various points in time also allows you to perform trend analysis, which enables you to manage your environment more efficiently and more proactively.

When you issue the REP command, a panel is displayed that allows you to specify the content of the report. You choose which types of objects that you want included in the report. For example, for a database, you might want a report that lists the table spaces, tables, and indexes in the database. Or, for a group of schemas, you might want a report that lists the distinct types in each schema.

After you specify the objects for the report, DB2 Admin generates JCL for a batch job that produces the report in a printable format. The batch job contains two steps. The first step invokes the GEN function to produce a version file for the objects that are to be included in the report. The second step formats the records in the version file into a report that is written to a data set.

The generated report consists of the following sections:
- A summary section that lists which types of objects are included in the report (the GEN parameters that were active when the data was collected).
- A detailed report section for each type of object that is included in the report. Each detailed report section lists all of the occurrences of the particular object. The information that is provided for each object and the column headings are the same as what is displayed on the corresponding system catalog panel for the object.

The following figure shows an example of the summary section of the report:

```
ADB2GEN parameters active when this data was collected :

Create Database(s)     : Yes   Create Tablespace(s)   : Yes   Create Table(s)        : Yes
Create View(s)         : No    Create Index(es)       : No    Create Synonym(s)      : No
Create Alias(es)       : No    Create Label(s)        : No
Create Triggers        : No    also for refs not gen'd : No
Create Foreign key(s)  : No    also for refs not gen'd : No
Create User def. Types : No    Create Functions       : No    Create Stored Procedures: No

Column information will not be included in this report.
```

Figure 73. Example of the summary section

The following figure shows an example of a detailed report section for table spaces:

```
Name      DB Name    Parts Bpool  L E S I C Tables  Act. pages  Segsz T L
*         *              * *      * * * * *      *           *      * * *
-------- -------- ------ ------ - - - - - ------ ----------- ------ - -
SYSALTER DSNDB06       0 BP32K  P N A N N      2          44      4   Y
SYSCOPY  DSNDB06       0 BP0    A N A N N      2         720      0   Y
SYSDBASE DSNDB06       0 BP8K0  A N A N N     14        8280      0   Y
SYSDBAUT DSNDB06       0 BP0    A N A N N      4          84      0   Y
SYSDDF   DSNDB06       0 BP0    P N A N N      8          38      4   Y
SYSEBCDC DSNDB06       0 BP0    P N A N N      1          12      4   Y
SYSGPAUT DSNDB06       0 BP0    A N A N N      1         720      0   Y
SYSGROUP DSNDB06       0 BP0    A N A N N      2          24      0   Y
SYSGRTNS DSNDB06       0 BP8K0  R N A N N      2          24      4   Y
SYSHIST  DSNDB06       0 BP8K0  R N A N N      9         144      4   Y
SYSJAUXA DSNDB06       0 BP0    L N A N N      1         288      0 0 Y
SYSJAUXB DSNDB06       0 BP0    L N A N N      1        1008      0 0 Y
```

Figure 74. Example of a detailed report section - table spaces

Generating a report

You can generate reports that can be saved and printed.

About this task

To generate a report:

Procedure

1. From the DB2 Admin Main Menu, specify option 1 to display the System Catalog panel.
2. Select one of the options that supports the ability to specify the REP command to generate a report. The following options support the ability to specify the REP command to generate a report:
   - Databases (option D)
   - Table Spaces (option S)
   - Tables, Views, and Aliases (option T)
   - Aliases (option A)
   - Synonyms (option Y)
   - Schemas (option H)
   - Data (or Distinct) Types (option E)
   - Functions (option F)
   - Storage Groups (option G)
   - Stored Procedures (option O)
   - Triggers (option J)
   - Sequences (option Q)

   You can use the fields at the bottom of the panel to specify search criteria to filter or limit the number of objects that are displayed.

3. Generate a report.
   - To generate a report for the single object, issue the REP line command.
   - To generate a report for all of the listed objects, issue the REP primary command.

   The following figure shows the REP command being issued against a database.

```
 DB2 Admin ------------------ DB2X Databases --------------------- Row 1 of 4
 Command ===>                                              Scroll ===> PAGE

 Commands: GRANT  MIG  DIS  STA  STO  UTIL
 Line commands:
  T - Tables  S - Table spaces  X - Indexes  G - Storage group  ICS - IC status
  DIS - Display database  STA - Start database  STO - Stop database  A - Auth
  ? - Show all line commands
                         Storage  Buffer        Created       Index
 Select Name     Owner   Group    Pool      DBID By      T E BPool    I
        *        *       *        *           * *       * * *        *
 ------ -------- -------- -------- -------- ------ -------- - - -------- -
        DSNATPDB DB2ADM   SYSDEFLT BP0         260 ISTJ      E BP2      Y
        DSNDB04  SYSIBM   SYSDEFLT BP0           4 SYSIBM      BP0      N
 REP    DSNDB06  SYSIBM   SYSDEFLT BP0           6 SYSIBM    E BP0      N
        DSN8D81A DB2ADM   DSN8G810 BP0         258 ISTJ      E BP2      Y
        DSN8D81P DB2ADM   DSN8G810 BP0         259 ISTJ M    E BP2      Y
 ****************************** END OF DB2 DATA ******************************
```

Figure 75. Databases panel (ADB21D) - Example of issuing the REP command to generate a report

When you use either the REP line command or the REP primary command, the Generate Report from DB2 Catalog panel, as shown in the following figure, is displayed. The Generate Report from DB2 Catalog panel that is displayed (ADB2REP, ADB2REPS, or ADB2REP6) and the fields that are included on the panel depend on the type of object that the REP command was issued for.

4. Fill in the fields on the Generate Report from DB2 catalog panel and press Enter.
5. Use the panel to specify the following items:
   - The types objects that you want included in the report. Specify Y for each object type that you want. For triggers, specify D to include triggers that refer to extracted tables.
   - Whether to have the column properties for objects that have associated columns included in the report. This field is displayed only when it is applicable.
   - The data set information for the report output.

```
 DB2 Admin ------------ DB2X Generate Report from DB2 Catalog ------------ 13:24
 Option ===>

  Generate batch report for database DSNDB06              DB2 System: DB2X
                                                          DB2 SQL ID: ISTJ
                                                                   More:     +
 Object types to be included from the DB2 catalog:
   Database . . . . . . . . . Y (Y,N)
   Table space  . . . . . . . Y (Y,N)
   Table  . . . . . . . . . . Y (Y,N)
   View . . . . . . . . . . . N (Y,N)
   Index  . . . . . . . . . . N (Y,N)
   Synonym  . . . . . . . . . N (Y,N)
   Alias  . . . . . . . . . . N (Y,N)
   Trigger  . . . . . . . . . N (Y,N,D)
   Storage group  . . . . . . N (Y,N)
   Plan/package . . . . . . . N (Y,N)

 Include column data  . . . . Y (Y,N)

 Output file:
   Data set name  . . . . . . 'USER.DB0024.REPORT'                          >
     Data set disposition . . OLD   (OLD, SHR, or MOD)

 BP - Change batch job parameters
```

Figure 76. Generate Report from DB2 Catalog panel (ADB2REP)

The batch jobs to create the report are generated, and an ISPF Edit session is displayed.
6. Verify and submit the generated jobs. The report is created in a printed format and written to the data set that was specified.

Results

You are now ready to print the data set with the carriage control and specified rotate options.

Redefined columns in the DB2 catalog

DB2 Admin puts integers in the INTEGER column to improve readability.

In some DB2 catalog tables, when a column with an INTEGER data type became too small to hold large values, DB2 added a corresponding column with a FLOAT data type to the catalog table to replace the INTEGER column. For example, CARDF was added for CARD in SYSTABLES, and FIRSTKEYCARDF was added for FIRSTKEYCARD in SYSINDEX. The *DB2 SQL Reference* shows that the

INTEGER version of the column is no longer used. When the catalog table is queried by using SPUFI (SELECT *), the value for the unused column might be displayed as 0 or -1.

DB2 Admin handles these pairs of INTEGER and FLOAT columns differently than DB2 does. For readability, DB2 Admin displays the integer equivalent of the value that is in the FLOAT column in the INTEGER column if the value fits. If the value is too large, DB2 Admin displays 11 asterisks instead.

The following columns contain the corresponding integer value of the floating point column when the floating point column also is present in the SELECT list:

- CARD
- COLCARD
- FIRSTKEYCARD
- FULLKEYCARD
- FREQUENCY
- FAROFFPOS
- NEAROFFPOS
- NACTIVE
- NPAGES
- SPACE
- KEYCOUNT
- CLUSTERRATIO

In addition, the integer columns must be returned by DB2 as INTEGER NOT NULL.

Within a SELECT in DB2 Admin, use one of the following methods to get the real value from the DB2 catalog:

- Do not specify both the integer and floating point column in the SELECT list.
- Rename one of the columns in the SELECT list by using AS, for example, SELECT CARD AS MYCARD.
- Change the data type in the result, for example, SELECT DECIMAL(COLCARD,11,0)

DB2 Admin restrictions on DB2 object names

There are two DB2 Admin restrictions on DB2 object names.

DB2 Admin puts two restrictions on DB2 object names. Do not use:

- Object names that contain Unicode characters that cannot be translated into the EBCDIC CCSID that DB2 Admin is using.
- Object names that contain an apostrophe (').

Object names that contain these characters can be displayed, but when a line command is used with either of the restricted object names, an error (SQLCODE -104) or warning message might be displayed.

Chapter 9. Building and running SQL statements

DB2 Admin can issue, build, and run SQL statements.

This information describes how to use DB2 Admin to perform the following tasks:
- Issue dynamic SQL statements from your screen, from a data set, or from program file
- Build and run SQL SELECT, INSERT, UPDATE, and DELETE statements interactively by using line commands
- Run the following SQL statements by entering required parameters: CREATE, DROP, LABEL ON, COMMENT ON, GRANT, and REVOKE

  The two panels for this function are also used from the system catalog panels, where they are shown when a line command is issued against an object. When invoked in this way, the object names contain the object name from the catalog.

Topics:

Selecting a method for building and running SQL statements
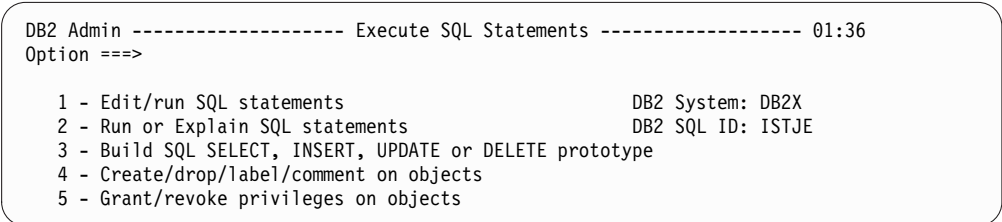
Use the Execute SQL Statements panel to choose how you want to build and run SQL statements.

About this task

To use the Execute SQL Statements panel to choose how you want to build and run SQL statements:

Procedure

1. Select option 2 on the Administration Menu panel. The Execute SQL Statements panel is displayed, as shown in the following figure,

```
DB2 Admin ------------------- Execute SQL Statements ------------------- 01:36
Option ===>

   1 - Edit/run SQL statements                      DB2 System: DB2X
   2 - Run or Explain SQL statements                DB2 SQL ID: ISTJE
   3 - Build SQL SELECT, INSERT, UPDATE or DELETE prototype
   4 - Create/drop/label/comment on objects
   5 - Grant/revoke privileges on objects
```

Figure 77. Execute SQL Statements panel (ADB22)

2. Select one of the following options:

1 - Execute SQL statements from screen input
Select this option to run SQL statements from your screen.

2 - Run or Explain SQL statements
Select this option to run SQL statements from a data set or to run or explain an SQL statement from a program file. When you use a data set, you can edit the SQL statements by using the ISPF editor, save the edited statements, and run the statements later. When you use a program file, you can select one SQL statement at a time to run or explain.

3 - Build SQL SELECT, INSERT, UPDATE or DELETE prototype
Select this option to build and run an SQL SELECT, INSERT, UPDATE or DELETE statement. The statement is built interactively using line commands.

4 - Create/drop/label/comment on objects
Select this option to run one of the following SQL statements: CREATE, DROP, LABEL ON, or COMMENT ON.

5 - Grant/revoke privileges on objects
Select this option to run GRANT and REVOKE SQL statements.

Running SQL statements from screen input

You can enter free-form SQL statements on your screen and run them.

About this task

To run SQL statements from screen input:

Procedure

1. Select option 1 on the Execute SQL Statements panel. The Edit/run SQL Statements panel is displayed, as shown in the following figure.

   Note: Lines preceding the statement that start with the SQL comment characters (--) are ignored.

```
DB2 Admin ------------ Edit/Run SQL Statement ------------ Columns 00001 00072
Command ===>                                                Scroll ===> CSR

****** ***************************** Top of Data ******************************
==MSG> Use command EXEC to run the SQL statement and return to the edit session
==MSG> or use command CANCEL to exit the edit session without running the SQL
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
****** ***************************** Bottom of Data ***************************
```

Figure 78. Edit/run SQL Statement panel (ADB221)

2. Enter the SQL statement you want to run between column 1 and 72 using the regular ISPF Edit commands. Line numbers should not be used.
3. Take one of the following actions:

- If you use END (PF3), the statement is saved in the temporary data set and the SQL statement is run.
- If you use END without any changes to the SQL statement, a prompt panel is displayed where you can specify whether the statement should be run or not.
- If you use the CANCEL command, you leave the edit panel without saving or running the SQL statement.

Results

If an SQL SELECT statement returns rows, the result is shown on the default table display panel.

You can edit an SQL statement by entering EDIT on the command line.

By default, any SQL statement that you enter is converted to uppercase. To disable this, use the CAPS OFF primary command.

What to do next

To run the SQL statement and return to the edit panel, you can enter the EXEC command from the editor primary command line.

Running SQL statements from a data set

You can run SQL statements that are stored in a data set.

About this task

To run SQL statements that are stored in a data set:

Procedure

1. Select option 2 on the Execute SQL Statements panel. The Run or Explain SQL Statements panel is displayed, as shown in the following figure.

```
DB2 Admin ---------------- Run or Explain SQL Statements ---------------- 17:44
Option ===>


   1 - Run SQL statements from a data set              DB2 System: DB2X
        EDIT first ===> YES (Yes/No)                   DB2 SQL ID: ISTJE
   2 - Run or Explain SQL located in a program
        Program type ===>   (1=COBOL, 2=PL/I)

ISPF library:
   Project ===>
   Group   ===>           ===>           ===>           ===>
   Type    ===>
   Member  ===>                     (blank for member selection list)

Other partitioned or sequential data set:
   Data Set Name  ===>
   Volume Serial  ===>               (if not cataloged)

Alternative pre-allocated DD name:
   DD name ===>                      (use ddname(member) for members)
```

Figure 79. Run or Explain SQL Statements panel (ADB222)

2. Select option 1 to run the SQL statements from a data set.

3. Specify the data set name that contains the SQL statements that you want to run. The input data set can be specified as:
   - An ISPF library
   - A partitioned or sequential data set
   - A pre-allocated ddname

   Restriction: The following restrictions apply to the input data set you specify:
   - If the record format (RECFM) is either F or FB and the logical record length (LRECL) is either 79 or 80, DB2 Admin assumes that the last 8 bytes of each record are for sequence numbers. Therefore, you should not use the last 8 columns of each record to store SQL statements.
   - If the record format (RECFM) is either F or F and the logical record length (LRECL) is neither 79 nor 80, DB2 Admin assumes that all of the columns of each record are for SQL statements.
   - If the record format (RECFM) is either V or VB, DB2 Admin checks to see if the content in columns 1 through 8 of the first record is numeric. If it is, DB2 Admin assumes that the first 8 bytes of each record are for sequence numbers. If it is not, DB2 Admin assumes that all columns are for SQL statements.
4. Run the SQL statement.
   - If you specify `Yes` in the **EDIT first** field and press Enter, the SQL statements are placed in ISPF edit mode on the specified data set before running them. You can then edit the statements. Press End in the edit session to run the SQL statements.
   - If you specify `No` in the **EDIT first** field, press Enter to run the SQL statements.

Running or explaining SQL statements from a program file

You can run or explain SQL statements that are in a program file.

About this task

To run or explain SQL statements that are in a program file:

Procedure

1. Select option 2 on the Execute SQL Statements panel The Run or Explain SQL Statements panel is displayed, as shown in the following figure.

```
DB2 Admin ---------------- Run or Explain SQL Statements ---------------- 17:44
Option ===>


  1 - Run SQL statements from a data set            DB2 System: DB2X
        EDIT first ===>     (Yes/No)                DB2 SQL ID: ISTJE
  2 - Run or Explain SQL located in a program
        Program type ===> 1 (1=COBOL, 2=PL/I)

ISPF library:
  Project ===>
  Group   ===>           ===>           ===>           ===>
  Type    ===>
  Member  ===>                      (blank for member selection list)

Other partitioned or sequential data set:
  Data Set Name  ===>
  Volume Serial  ===>                (if not cataloged)

Alternative pre-allocated DD name:
  DD name ===>                       (use ddname(member) for members)
```

Figure 80. Execute SQL Statements from a Data Set panel (ADB222)

2. Select option 2 to specify that the SQL statements to run or explain are in a program file and specify the type of program. The types are:
   1   COBOL
   2   PL/I

   If you specify the program type as a parameter when you issue the RUN or EXPLAIN primary command for the SQL statement, the parameter for the type overrides the value that is set in the **Program type** field.

3. Specify the data set name that contains the program. The input data set can be specified as:
   - An ISPF library
   - A partitioned or sequential data set
   - A pre-allocated ddname

4. Press Enter to display the program file, as shown in the following figure.

```
ISREDDE2   ELACZ.TEST.SQLSTMT(TEST) - 01.16          Columns 00001 00080
Command ===>                                         Scroll ===> PAGE
****** ****************************** Top of Data *********************************
==MSG>
==MSG> Use the line command "C" or block command "CC" to select
==MSG> an SQL statement.
==MSG> Use the primary command "EXPLAIN" to explain or "RUN" to run
==MSG> the selected SQL statement.
==MSG>
000001 ------------------------------------------------------------------------
:
000010  --
000011  EXEC SQL
000012     SELECT NAME,
000013            TBNAME,
000014            TBCREATOR,
000015            COLNO,
000016            COLTYPE,
000017     FROM SYSIBM.SYSCOLUMNS
000018     WHERE TBNAME = :TBN
000019       AND TBCREATOR = :TBC
000020     ORDER BY NAME, TBNAME;
****** ***************************** Bottom of Data *******************************
```

Figure 81. Example of selecting SQL statements in a program to run or explain

5. Use the C line command or the CC block command to select the SQL statement to run or explain. Only one SQL statement can be selected at a time.

   Restriction: The following SQL statements cannot be run or explained:
   - ALLOCATE CURSOR
   - ASSOCIATE LOCATOR
   - BEGIN DECLARE SECTION and END DECLARE SECTION
   - CALL
   - CLOSE
   - CONNECT
   - DECLARE STATEMENT, DECLARE TABLE, DECLARE VARIABLE
   - all DESCRIBE statements
   - EXECUTE and EXECUTE IMMEDIATE
   - FETCH
   - FREE LOCATOR and HOLD LOCATOR
   - INCLUDE
   - OPEN
   - PREPARE
   - SIGNAL SQLSTATE
   - VALUES
   - WHENEVER
   - --#SET ROWS_FETCH, --#SET ROWS_OUT, --#SET TERMINATOR

6. Issue the RUN primary command to run the statement or the EXPLAIN primary command to explain the statement.

7. Specify the values for every host variable in the SQL statement in the pop-window that is displayed. Enter the values for character host variables in single quotation marks. If you leave the value of host variable blank, the host variable is removed from the statement.

8. Exit the edit session to have the primary command executed.

   Tip: If you have changed the selected statement but do not want to save the changes in the program file, choose CANCEL when you are prompted to exit the edit session. The updated statement is executed, but the program file is not changed.

Building SQL SELECT, INSERT, UPDATE and DELETE prototypes

You can build SQL SELECT, INSERT, UPDATE and DELETE prototypes interactively by using DB2 Admin line commands.

About this task

Because prototyping is similar for each of the SQL statements covered by this option, this information describes only how to build the SELECT statement.

To build SQL SELECT, INSERT, UPDATE and DELETE prototypes:

Procedure

1. Select option 3 on the Execute SQL Statements panel to display the Build SQL SELECT, INSERT, UPDATE or DELETE Prototype panel. Use this panel to search for the object (table, view, or alias) for which you want to build and run an SQL SELECT, DELETE, INSERT, or UPDATE statement.

```
ADB223 in ------------ Build SQL Prototype: Search Objects ------------ 06:22
 Command ===> _____

 Enter/verify:
  Schema  . . . _____ >           (optional, default is SMITHJR)
  Name  . . . . _____ > (optional)
```

Figure 82. Build SQL SELECT Prototype panel (ADB223)

2. Enter the Schema or Name of the object.
3. Press Enter to display a list of objects that match the search criteria, as shown in the following figure.

```
ADB223T n ----------- DB2X Tables, Views, and Aliases -------- Row 1 to 7 of 7
Command ===>                                               Scroll ===> PAGE

Line commands:
 SEL - Select for SQL SELECT prototype  T - Table
 DEL - DELETE prototype  INS - INSERT prototype  UPD - UPDATE prototype

Select  Name              Schema   T
        *                 *        *
------- ----------------> -------- -
_____         AA1122            OWNER1   T
_____         AARVV1145600_ANDR OWNER1   T
_____         EEMP              DSN8810  T
_____         EEPA              DSN8810  T
__SEL__         EMP               DSN8810  T
_____         EMPPROJACT        DSN8810  T
_____         EPROJ             DSN8810  T
_____         EPROJACT          DSN8810  T
_____         MAP_TBL           DSN8810  T
_____         NEWDEPT           DSN8810  T
_____         NEWPHONE          DSN8810  T
_____         PARTS             DSN8810  T
_____         PROJ              DSN8810  T
_____         PROJACT           DSN8810  T
_____         STAFF             DSN881SA T
_____         STAFFV1           DSN881SA V
_____         TCONA             DSN8810  T
_____         TDSPTXT           DSN8810  T
_____         TESTSTUFF         DSN881SA T
_____         TOPTVAL           DSN8810  T
_____         VACT              DSN8810  V
_____         VASTRDE1          DSN8810  V
_____         VASTRDE2          DSN8810  V
_____         VCONA             DSN8810  V
_____         VDEPMG1           DSN8810  V
_____         VDEPT             DSN8810  V
_____         VDSPTXT           DSN8810  V
_____         VEMP              DSN8810  V
```

Figure 83. Example of building an SQL SELECT statement (part 1 of 5) (ADB223T)

4. Build your SQL statement by using line commands. For example, if you want to build an SQL SELECT statement that returns the name and department number of all employees with a salary greater than $30,000, begin by using the SEL line command to select the table that contains the desired information. The previous figure shows that the EMP is selected. When you press Enter, DB2 Admin displays the panel in the following figure, which shows the partially built SQL statement at the top.

```
ADB21TSE --------- DB2X Build SQL SELECT Prototype -------------- Row 1 of 14
Command ===>                                              Scroll ===> PAGE

SELECT ?
  FROM DSN8810.EMP T
  FOR?
  WHERE ?
ORDER BY ?
GROUP BY ?
Commands: EDIT  RESET  *  QUOTE  INS  UPD  DEL  COUNT  COUNT_BIG
Line commands:  S - Show  SA - Show ASC  SD - Show DESC
 AVG, COUNT, COUNT_BIG, MAX, MIN, STDDEV, SUM, VARIANCE - Aggregate functions
 <oper><expr>, OR <pred>, IN list, BETWEEN <expr>,<expr> - WHERE predicates
 ? - Show all line commands

 Select                         Column Name      Col Type     Length
                                *                *            *
 -------------------------------- ----------------- -------- -----------
                                EMPNO            CHAR             6
 S                              FIRSTNME         VARCHAR          12
 S                              MIDINIT          CHAR             1
 S                              LASTNAME         VARCHAR          15
 S                              WORKDEPT         CHAR             3
                                PHONENO          CHAR             4
                                HIREDATE         DATE             10
                                JOB              CHAR             8
                                EDLEVEL          SMALLINT         2
                                SEX              CHAR             1
                                BIRTHDATE        DATE             10
 >30000                         SALARY           DECIMAL          9
                                BONUS            DECIMAL          9
                                COMM             DECIMAL          9
****************************** END OF DB2 DATA ******************************
```

Figure 84. Example of building an SQL SELECT statement (part 2 of 5) (ADB21TSE)

The following primary commands are available:

EDIT
> Edit the query. Editing does not change the SQL statement on the panel.

RESET
> Reset the query.

* (asterisk)
> Show all columns in the result.

QUOTE
> Place column names in quotes.

INS
> Insert statement prototype. Not applicable to creating a view.

UPD
> Update statement prototype. Not applicable to creating a view.

DEL
> Delete statement prototype. Not applicable to creating a view.

COUNT(*)
> Count distinct for this column returns integer value.

COUNT_BIG(*)
> Count distinct for this column returns decimal value.

The following line commands are available:

S Show the column in the result.

SA Show the column in the result and sort ascending. Not applicable to creating a view.

SD Show the column in the result and sort descending. Not applicable to creating a view.

AVG
  Return average value for the numeric column.

COUNT
  Count distinct for this column returns integer value.

COUNT_BIG
  Count distinct for this column returns decimal value.

MAX
  Return maximum value for the numeric column.

MIN
  Return minimum value for the numeric column.

STDDEV
  Return the standard deviation for the numeric column.

SUM
  Returns the sum of the selected columns.

VARIANCE
  Return the variance of a set of numbers from selected columns.

The WHERE predicate can be:

<oper><expr>
  where:

  <oper>
    Adds a predicate (WHERE clause) for this column with this operator. <oper> can be: =, ¬=, >, >=, <, <=, or LIKE.

  <expr>
    Right side of predicate, consisting of an alphanumeric value.

OR <pred>,
  Examples:
  ```
  OR=10
  R=x
  OR IN(1,2,3,4,5)
  OR BETWEEN s,t
  ```

IN list
  Examples:
  ```
  IN x,y
  IN('x','y')
  IN 1,2,3,4,5,6
  ```

BETWEEN <expr>, <expr>
  Examples:
  ```
  BTW x,y
  BETWEEN x AND y
  BTW nnn,ppp
  ```

For this scenario, use the S line command to include columns in your SELECT statement, and use the <oper><expr> line command to specify the salary range of 30,000.

Press Enter to run the line commands and to update the SELECT statement, as shown in the following figure.

```
 DB2 Admin --------- DB2X Build SQL SELECT Prototype ------------- Row 1 of 14
 Command ===>                                                 Scroll ===> PAGE

 SELECT FIRSTNME,MIDINIT,LASTNAME,WORKDEPT,SALARY
   FROM DSN8810.EMP T
   FOR?
   WHERE SALARY>30000
 ORDER BY ?
 GROUP BY ?
Commands: EDIT  RESET  *  QUOTE  INS  UPD  DEL  COUNT  COUNT_BIG
Line commands:  S - Show  SA - Show ASC  SD - Show DESC
 AVG, COUNT, COUNT_BIG, MAX, MIN, STDDEV, SUM, VARIANCE - Aggregate functions
  <oper><expr>, OR <pred>, IN list, BETWEEN <expr>,<expr> - WHERE predicates
  ? - Show all line commands
Select                           Column Name      Col Type     Length
                                 *                *                 *
-------------------------------- ---------------- -------- -----------
                                 EMPNO            CHAR              6
 *S                              FIRSTNME         VARCHAR          12
 *S                              MIDINIT          CHAR              1
 *S                              LASTNAME         VARCHAR          15
 *S                              WORKDEPT         CHAR              3
                                 PHONENO          CHAR              4
                                 HIREDATE         DATE             10
                                 JOB              CHAR              8
                                 EDLEVEL          SMALLINT          2
                                 SEX              CHAR              1
                                 BIRTHDATE        DATE             10
 SD                              SALARY           DECIMAL           9
                                 BONUS            DECIMAL           9
                                 COMM             DECIMAL           9
****************************** END OF DB2 DATA ******************************
```

Figure 85. Example of building an SQL SELECT statement (part 3 of 5) (ADB21TSE)

Use the SD line command, as shown in the previous figure, to add the ORDER BY clause to the SELECT statement. When you press Enter, the SELECT statement is updated and displayed, as shown in the following figure.

```
 DB2 Admin --------- DB2X Build SQL SELECT Prototype -------------- Row 1 of 14
 Command ===>                                              Scroll ===> PAGE

 SELECT FIRSTNME,MIDINIT,LASTNAME,WORKDEPT,SALARY
   FROM DSN8810.EMP T
   FOR?
   WHERE SALARY>30000
ORDER BY SALARY DESC
GROUP BY ?
Commands: EDIT  RESET  *  QUOTE  INS  UPD  DEL  COUNT  COUNT_BIG
Line commands:  S - Show  SA - Show ASC  SD - Show DESC
 AVG, COUNT, COUNT_BIG, MAX, MIN, STDDEV, SUM, VARIANCE - Aggregate functions
  <oper><expr>, OR <pred>, IN list, BETWEEN <expr>,<expr> - WHERE predicates
  ? - Show all line commands
Select                            Column Name      Col Type     Length
                                  *                *               *
-------------------------------- ------------------ -------- -----------
                                  EMPNO             CHAR             6
                                  FIRSTNME          VARCHAR         12
                                  MIDINIT           CHAR             1
                                  LASTNAME          VARCHAR         15
                                  WORKDEPT          CHAR             3
                                  PHONENO           CHAR             4
                                  HIREDATE          DATE            10
                                  JOB               CHAR             8
                                  EDLEVEL           SMALLINT         2
                                  SEX               CHAR             1
                                  BIRTHDATE         DATE            10
 *SD                              SALARY            DECIMAL          9
                                  BONUS             DECIMAL          9
                                  COMM              DECIMAL          9
****************************** END OF DB2 DATA ******************************
```

Figure 86. Example of building an SQL SELECT statement (part 4 of 5) (ADB21TSE)

The SQL statement is now ready to be run. Do not specify any line commands when running the statement. When you press Enter, the result is displayed, as shown in the following figure.

```
 DB2 Admin -------------- DB2 Result of the SQL SELECT ------------- Row 1 of 8
 Command ===>                                              Scroll ===> PAGE

L FIRSTNME    MIDINIT LASTNAME        WORKDEPT     SALARY
  *           *       *               *                *
- ----------- ------- --------------- -------- -----------
  CHRISTINE   I       HAAS            A00         52750.00
  DIAN        J       HEMMINGER       A00         46500.00
  VINCENZO    G       LUCCHESI        A00         46500.00
  MICHAEL     L       THOMPSON        B01         41250.00
  JOHN        B       GEYER           E01         40175.00
  SALLY       A       KWAN            C01         38250.00
  EVA         D       PULASKI         D21         36170.00
  IRVING      F       STERN           D11         32250.00
****************************** END OF DB2 DATA ******************************
```

Figure 87. Example of building an SQL SELECT statement (part 5 of 5) (ADB2DF)

You can also perform SQL prototyping by using the Create View panel (ADB26CV), as shown in the following figure.

You can use the EDIT command to capture the SELECT statement and store it in a data set.

```
DB2 Admin --------------------- DB2X Create View --------------------- 10:4
Command ===>

 CREATE VIEW
 Owner     ===>        >              (optional, default is ISTJE)
 Name      ===>                     > (? to look up)
 (                                    (optional column list)
 Col names  ===>

 ) AS                               (? to use SELECT prototype)
 SELECT stmt===>




 WITH ? CHECK OPTION    (check INSERTS/UPDATES with VIEW definition)
 Check opt  ===>        (Y-include, N-omit, blank-omit (default),
                         C-CASCADED, L-LOCAL)
```
Figure 88. SQL prototyping on the Create View panel (ADB26CV)

Issuing CREATE, DROP, LABEL ON, and COMMENT ON statements

Use the Execute SQL Statements panel to issue CREATE, DROP, LABEL ON, and COMMENT ON statements.

Select option 4 on the Execute SQL Statements panel to display the Create/Drop/Label/Comment On Objects panel, as shown in the following figure.

Use this panel to issue CREATE, DROP, LABEL ON, and COMMENT ON statements.

Restriction: When creating SQL scalar functions, the maximum length of the return statement is 2MB (32,767KB). When creating SQL stored procedures, the maximum length of the procedure body is 2MB (32,767KB).

```
ADB26 min ---------- DSNB Create/Drop/Label/Comment On Objects ---------- 04:35
 Option ===> _____


                                               DB2 System: DSNB
 CREATE                                DROP          DB2 SQL ID: ULVEMAN
   CG  - Storage group                  DG  - Storage group
   CD  - Database                       DD  - Database
   CS  - Table space                    DS  - Table space
   CT  - Table                          DT  - Table
   CV  - View                           DV  - View
   CL  - Alias for table                DL  - Alias
   CLQ - Alias for sequence
   CX  - Index                          DX  - Index
   CY  - Synonym                        DY  - Synonym
   CA  - Auxiliary table
   CE  - User-defined type              DE  - User-defined type
   CJ  - Trigger                        DJ  - Trigger
   CF  - Function                       DF  - Function
   CO  - Stored procedure               DO  - Stored procedure
   CM  - Materialized table (MQT)
   CQ  - Sequence                       DQ  - Sequence
  CGV  - Global variable               DGV  - Global variable
  CTR  - Trusted context               DTR  - Trusted context
  CRO  - Role                          DRO  - Role
  CCM  - Column mask                   DCM  - Column mask
  CPM  - Row permission                DPM  - Row permission
 LABEL                                COMMENT (remark)
   LT  - Table/view                     RT  - Table/view
   LL  - Alias                          RL  - Alias
   LC  - Column                         RC  - Column
                                        RE  - Distinct type
                                        RF  - Function
                                        RO  - Stored procedure
                                        RJ  - Trigger
                                        RX  - Index
                                        RQ  - Sequence
                                       RGV  - Global variable
                                       RTR  - Trusted context
                                       RRO  - Role
                                       RCM  - Column mask
                                       RPM  - Row permission
```

Figure 89. Create/Drop/Label/Comment On Objects panel (ADB26)

You can use this panel to perform the following tasks:
- Create a database
- Create a table space
- Create a table
- Create a materialized query table
- Create an index
- Place a label on a table
- Comment on a table
- Drop a table and use Drop Impact Reports
- Create, drop, or comment on a trusted context
- Create, drop, or comment on a role
- Create, drop, or comment on column masks and row permissions.
- Create a global variable

You can use the following examples as models when using panel ADB26 to create other objects.

Creating a database

Use the Create Database panel to create a new database.

About this task

To create a new database:

Procedure

1. Select option CD on the Create/Drop/Label/Comment On Objects panel. The Create Database panel is displayed, as shown in the following figure.

```
DB2 Admin -------------------- DSNB Create Database -------------------- 09:25
Command ===>

 CREATE

 DATABASE . . . . .              (required database name. ? to look up existing)

 STOGROUP . . . . .               > (optional: default is SYSDEFLT. ? to look up)

 BUFFERPOOL . . . .              (optional: default is defined during installation)

 INDEXBP  . . . . .              (optional: default is defined during installation)

 CCSID  . . . . . .              (optional: ASCII/EBCDIC/UNICODE)

 AS WORKFILE  . . .              (Yes/No, only for data sharing environments)


   FOR MEMBER . . .              (optional: default is current connected member)
```

Figure 90. The Create Database panel (ADB26CD)

2. Specify the following values:
   - In the **DATABASE** field, enter a database name for the new database, or enter a question mark (?) to look up existing database names using the Databases panel.
   - Optional: In the **STOGROUP** field, specify the name of a storage group in which you want the new database to belong.\
   - Optional: In the **BUFFERPOOL** and **INDEXBP** fields, enter the names of buffer pools to use (as defined at installation time).
   - Optional: In the **CCSID** field, enter one of the following formats: ASCII, EBCDIC, or UNICODE.
   - Optional: In the **AS WORKFILE** field, enter a work file name for data sharing environments.
3. In the **AS TEMP** field, specify whether to create a database for declared temporary tables.
4. In the **FOR MEMBER** field, specify a different member in which to place the new database.
5. Follow the instructions on the Statement Execution Prompt panel (if enabled), as shown in the following figure, to complete and run the SQL statement for creating the new database.

```
DB2 Admin --------------- DSNX Statement Execution Prompt --------------- 18:10
Option ===>

DB2 Admin is about to execute the statement below. You have asked to be
prompted before DB2 Admin executes this type of statement. What do you want to
do now:
   1 - Execute the statement
   2 - Edit the statement
   3 - Create a batch job with the statement
   4 - Add the statement to the work statement list
 CAN - Cancel
 Work statement list dsn  ===>
 Work statement list name ===>              Action ===> A (Append or Replace)
                                                          More:     +
Statement that is about to be executed (first 28 lines):
CREATE DATABASE "DBTEST01"
```

Figure 91. The Statement Execution Prompt panel (ADB2PSTM) – Creating a new database

Creating a table space

Use the Create Table Space panel to create a new table space in a database.

About this task

To create a new table space in a database:

Procedure

1. Select option CS on the Create/Drop/Label/Comment On Objects panel. The
   Create Table Space panel is displayed, as shown in the following figure.

```
ADB26CS n -------------------DB2X Create Table Space ------------------- 06:28
Command ===> _____

 CREATE

 TABLESPACE . . FGRTS    (required table space name.  ? to look up)

 IN . . . . . . FGRDB    (optional database. default=DSNDB04. ? to look up)

 Like:
 Database . . . _____  (optional existing database. ? to look up)
 Name . . . . . _____  (optional existing table space. ? to look up)
```

Figure 92. The Create Table Space panel (ADB26CS)

2. Specify the following values:
   * In the **TABLESPACE** field, enter a table space name for the new table space,
     or enter a question mark (?) to look up existing table space names using the
     Table Spaces panel.
   * Optional: In the **IN** field, specify the name of a database in which you want
     the new table space created, or enter a question mark (?) to look up existing
     database names using the Databases panel.
   * Optional: In the **Like: Database** field, enter the name of a database on which
     to model the new table space.
   * Optional: In the **Like: Name** field, enter a table space name on which to
     model the new table space.

3. Press Enter to display the Create Table Space panel, as shown in the following figure.

```
ADB21SAR  ----------------- DSN9 Create Table Space --------- Row 1 to 1 of 1
Command ===> _____  Scroll ===> PAGE

Commands: NEXT  ORIGINAL                      MAKEPBG  MAKEPBR  MAKEPBR2
Line commands: I - Insert part  D - Delete part  U - Update part
              C - Clear data  R - Repeat part
CREATE TABLESPACE: FGRTS    IN  FGRDB

Numparts  . . . . . 0         Large . . . . . ___    LOB . . . . . ___
Define  . . . . . . ___       DSSIZE  . . . . ___    LOG . . . . .
Member Cluster  . . ___       SEGSIZE . . . . __     CCSID . . . .
Buffer Pool . . . . _____   Close Rule  . . ___    Max Rows  . . __
Lock Size . . . . . _____  Lock Part . . .        Lock Max  . . _____
Max Partitions  . . 0_____ PAGENUM . . . .
                              Free  Pct        E T S
S   Part      Pqty      Sqty Page Free Compr R M T VCAT    Stogroup GBPCache
- ------ ---------- ---------- ---- ---- ----- - - - -------- ------->  --------
_      0  _____  _____   ___   _ ___   _ _ _ _____  _____  _____
```

Figure 93. The Create Table Space panel (ADB21SAR) – Creating a new table space

4. On the Create Table Space panel (ADB21SAR), specify parameters for the new table space or issue the CONTINUE primary command to use the default settings.
5. Follow the instructions on the Statement Execution Prompt panel (if enabled) to complete and run the SQL statement for creating the new table space.

Creating a table

Use the Create Table panel to create a new table.

About this task

To create a new table in a table space within a database:

Procedure

1. Select option CT on the Create/Drop/Label/Comment On Objects panel. The Create Table panel is displayed, as shown in the following figure.

```
ADB26CT n ---------------------DB2X Create Table ---------------------- 16:31
Command ===>

 CREATE TABLE

 Schema  . . . . .         >          (default is )
 Name  . . . . . . NEW                > (? to look up)

 LIKE
 Schema  . . . . .         >
 Name  . . . . . .                    > (? to look up)
 Identity attrs  .          (Yes/No)
 Row chg attrs . .          (Yes/No)
 As model only . .          (Yes/No)

 (
 Number of columns . . . 6
```

Figure 94. The Create Table panel (ADB26CT)

2. Specify the following values:

- In the **Schema** field, enter the schema for the new table or use the default schema.
- In the **Name** field, enter a table name for the new table, or enter a question mark (?) to look up existing table names using the Tables, Views, and Aliases panel.
- Optional: In the **LIKE Schema** field, specify the schema on which to model the new schema for the new table.
- Optional: In the **LIKE Name** field, enter the name of a table on which to model the new table, or enter a question mark (?) to look up existing table names using the Tables, Views, and Aliases panel.
- In the **Identity attrs** field, specify whether to include identity column attributes for the new table.
- In the **Row chg attrs** field, specify whether to include row change timestamp attributes for the new table.
-
  Optional: In the **As model only** field, specify Y to indicate that you want to use the LIKE table as a model that you can edit before creating the table.
- Specify the number of columns for the table. In the panel in the previous figure, six columns are specified.

3. Press Enter to continue to the next Create Table Columns panel, as shown in the following figure.

```
ADB26CTF  ----------------- DSNA Create Table Columns ------- Row 1 to 3 of 3
Command ===>                                            Scroll ===> CSR

Schema  . .         >           Database  . . .
Name  . . . NEWTABLE         > Table space . .

Commands : CREATE  PRIMKEY  TBLOPTS  PART  HASH
Line commands:   M - Move  A - After  B - Before
 Inn - Insert  U - Update  D - Delete  Rnn - Repeat
 UM - Update XML modifiers
                                                          Operation
Select Column Name        Col Type  Length Scale  Null D Col No Type
       *                  *              *     * *      *      * *
------ ----------------->  --------- ------ ------ ---- - ------ ---------

*      T1                 TIMESTMP      13     11 N    N      1 UPDATE
*      T2                 TIMESTZ       15     11 N    N      2 UPDATE
*      T3                 TIMESTZ       12      6 N    N      3 UPDATE
*      T4                 DATE           4      0 N    N      4 UPDATE
*      T5                 INTEGER        4      0 N    N      5 UPDATE
*      T6                 DATE           4      0 N    N      6 UPDATE
****************************** END OF DB2 DATA ******************************
```

Figure 95. The Create Table panel (ADB26CTF) – Creating a new table

4. On the Create Table panel (ADB26CTF), specify parameters for the new table.
5. Follow the instructions on the Statement Execution Prompt panel (if enabled) to complete and run the SQL statement for creating the new table.

Creating a materialized query table

Use the Create Materialized Table panel to create a new materialized query table.

About this task

To create a new materialized query table in a table space within a database:

Procedure

1. Select option CM on the Create/Drop/Label/Comment On Objects panel. The Create Materialized Table panel is displayed, as shown in the following figure.

```
DB2 Admin --------------- DB2X Create Materialized Table --------------- 16:17
Command ===>

CREATE TABLE (Materialized)
Owner        ===> ISTJE    >          Database    ===> TESTDB  (? look up))
Name         ===> MTABLE01         >  Table space ===> SPACE01  (? look up))
Source Owner ===> OWNER1    >
Source Name  ===> TABLE1           >  (? look up)
(                                  (column list ? to look up)
Col names  ===> ?
                                                                          >
) AS                               (? to use SELECT prototype)
SELECT stmt===> ?




MAINTAINED BY SYSTEM/USER ===> S      (S-SYSTEM, U-USER, default SYSTEM)
ENABLE QUERY OPTIMIZATION ===> YES    (Yes/No, default YES)
DEFINITION ONLY                       (Will create base table only)
 IDENTITY COL ATTRIBUTES  ===> YES    (EXCLUDE, Yes/No, default NO)
 COLUMN DEFAULTS          ===> NO     (EXCLUDE, Yes/No, default NO)
```

Figure 96. The Create Materialized Table panel (ADB26CM)

2. Specify the following values:
   * In the **Owner** field, enter a table owner name for the new materialized query table.
   * In the **Name** field, enter a table name for the new materialized query table.
   * In the **Source Owner** field, enter the name of the owner of that source table.
   * In the **Source Name** field, enter a source table name on which the new materialized query table is based, or enter a question mark (?) to look up existing table names.
   * In the **Col names** field, enter the column names to be added to the new materialized query table, or enter a question mark (?) to look up existing column names.
   * In the **SELECT stmt** field, enter an SQL SELECT statement to build the materialized query table, or enter a question mark (?) to use the Build SQL SELECT Prototype panel to build one.
   * In the **MAINTAINED BY SYSTEM/USER** field, enter S if you want the DB2 system to update and maintain the table, or enter U if you want a user program to update and maintain the table.
   * In the **ENABLE QUERY OPTIMIZATION** field, enter Yes or No to use the DB2 query optimizer.
   * In the DEFINITION ONLY area, specify whether you want to exclude either or both of the following:

     IDENTITY COL ATTRIBUTES
     > Specify Yes to exclude identity column attributes, or specify No to include them.

     COLUMN DEFAULTS
     > Specify Yes to exclude column defaults, or specify No to include them.

If you specify Yes for both fields in the DEFINITION ONLY area, DB2 Admin creates a regular base table (type "T"), as opposed to a materialized query table.

3. Press Enter.
4. Follow the instructions on the Statement Execution Prompt panel (if enabled) to complete and run the SQL statement for creating the new materialized query table.

Creating an index on a table

Use the Create Index panel to create a new index on a table.

About this task

Using DB2 Admin, you can create a new index on a table in several ways:
- Select option CX on the Create/Drop/Label/Comment On Objects panel (ADB26).
- Use the CREX line command on the Tables, Views, and Aliases panel (ADB21T).
- Use the CRE line command on the Indexes panel (ADB21X).
- Select option CX on the Explain panel (ADB2E).

Each of these methods display the create index panels, beginning with the Create Index panel (ADB26CX).

To create a new index on a table:

Procedure

1. Select option CX on the Create/Drop/Label/Comment On Objects panel. The Create Index panel is displayed, as shown in the following figure.

```
ADB26CX n --------------------- DSNA Create Index --------------------- 16:17
Command ===>


 CREATE INDEX

 Schema  . . . .          >          (default is RIVERAF)
 Name  . . . . . IXFGRNEW            >  (? to look up)

 ON
 Table Schema  .         >           (default is RIVERAF)
 Table name  . . TBFGR              >  (? to look up)

 Partitions  . . 0                   (0 for nonpartitioned INDEX)

 Like:
 Index Schema  .         >           (required for Like usage)
 Index name  . .                    >  (? to look up)
```

Figure 97. The Create Index panel (ADB26CX)

2. Specify the following values:
   - In the **Owner** field, enter the name of the index owner for the new index or use the default owner.
   - In the **Name** field, enter an index name for the new index, or enter a question mark (?) to look up existing index names using the Indexes panel.
   - Optional: In the **Table owner** field, enter the name of the owner for a table name on which the index is based.

- In the **Table name** field, enter a table name or enter a question mark (?) to look up existing table names using the Tables, Views, and Aliases panel.
- Optional: In the **Partitions** field, enter the number of partitions for a partitioned index. For releases of DB2 prior to Version 8, you can specify up to 254 partitions. For DB2 Version 8, you can specify up to 4096 partitions.
- Optional: In the **Like: Index owner** field, specify the name of an owner on which to model the new owner for the new index.
- Optional: In the **Like: Index name** field, enter the name of an index on which to model the new index or enter a question mark (?) to look up existing index names using the Indexes panel.

When you press Enter, the next Create Index panel (ADB21XAR) is displayed, as shown in the following figure.

```
ADB21XAR  -------------- DSNA Create Index ----------------- Row 1 to 2 of 2
Command ===>                                          Scroll ===> CSR

Commands: NEXT  ORIGINAL  EXPRESSION

Line commands: nnn A|D - Sequence & order  R - Remove the column  I - Include
 A - Ascending  D - Descending  RA - Random   U - Update expression/XML pattern
 B - Business Time without overlaps

CREATE INDEX RIVERAF    . IXFGRNEW            >
        ON RIVERAF.TBFGR
Unique . . . . . . .      Where Not Null . . .     Cluster . . . . .
Buffer Pool  . . . .      Close Rule . . . . .     Copy Allowed  . .
Piece Size . . . . .      Define . . . . . . .     Defer . . . . . .
Partitioned  . . . .      Padded . . . . . . .     Compress  . . . .
Exclude Null Keys  .

Select Column Name      Col Type     Length  Scale N ColSeq Ord
       *                *                *      * *      * *
------ ------------------ -------- ----------- ------ - ------ ---
       AXX               INTEGER         4      0 N    1 A
       BXX               CHAR            3      0 Y
***************************** END OF DB2 DATA ******************************
```

Figure 98. The Create Index panel (ADB21XAR)

3. On the upper portion of the Create Index panel (ADB21XAR), specify the index attributes.
   a. Specify whether the columns should be in ascending or descending order.
   b. Specify the general index attributes in the fields.

      Remember: Depending upon the version of DB2 that you are using and your choice of parameters, some of the attribute fields might be unavailable.

      Tip: Without negatively impacting query performance, you can improve the insert performance of NULL entries, by excluding NULL rows from an index. Type Yes in the **Exclude Null Keys** field to exclude NULL rows from a new index. The default is to include NULL keys in a new index.
4. On the scrollable table of the panel, use line commands to specify the columns in the index. All columns of the DB2 table are displayed. Index columns are identified in the Colseq and Order columns.
5. Issue the NEXT primary command to display the Create Index – Space panel (ADB21XAS).
6. Specify the space allocation and storage parameters for the index or for each partition of the index. If a partitioned index with more than one partition is being created, the word Default: appears at the beginning of the scrollable

portion of the panel. Use this line to enter common values for each partition and to avoid entering the same value for a parameter on all partitions again.

7. Issue the NEXT primary command to generate the DDL for the index and display an edit session.
8. Edit the CREATE statement or exit the session to create the index.
9. Follow the instructions on the Statement Execution Prompt panel (if enabled) to complete and run the SQL statement for creating the new index.

Results

After the index is created, DB2 Admin displays the Create Index – Utilities panel (ADB26CXU), on which you can run several index utilities, including RECOVER and RUNSTATS.

Placing a label on a table

Use the Label Table panel to place a label on a table.

About this task

To place a label on a table:

Procedure

1. Select option LT on the Create/Drop/Label/Comment On Objects panel. The Label Table panel is displayed, as shown in the following figure.

```
DB2 Admin -------------------- DB2X Label Table ----------------------- 15:34
Command ===>

 Owner    ===> OWNER1   >
 Name     ===> TABLE10           >  (? to look up)

 IS

 Label    ===>
```

Figure 99. The Label Table panel (ADB26LT)

2. Specify the following values:
   - In the **Owner** field, enter an owner name for the table on which you want to place a label.
   - In the **Name** field, enter a table name on which you want to place a label, or enter a question mark (?) to look up existing table names using the Tables, Views, and Aliases panel.
3. Enter a label for the table and press Enter.
4. Follow the instructions on the Statement Execution Prompt panel (if enabled) to complete and run the SQL statement for placing the label on the table.

Placing a comment on a table

Use the Comment Table panel to place a comment on a table.

About this task

To place a comment (or remark) on a table:

Procedure

1. Select option RT on the Create/Drop/Label/Comment On Objects panel. The Comment Table panel is displayed, as shown in the following figure.

```
DB2 Admin -------------------- DB2X Comment Table ---------------------- 15:43
 Command ===>

  COMMENT ON

  Schema  . . . D123         >
  Name  . . . . TABLE10           > (? to look up existing )



  IS

  Remarks . . .
```

Figure 100. The Comment Table panel (ADB26RT)

2. Specify the following values:
   - In the **Schema** field, specify a schema that is described in the catalog. Indicates a comment will be added or replaced for a schema.
   - In the **Name** field, enter a table name on which you want to place a comment or enter a question mark (?) to look up existing table names using the Tables, Views, and Aliases panel.
3. Enter a comment or remark for the table and press Enter.
4. Follow the instructions on the Statement Execution Prompt panel (if enabled) to complete and run the SQL statement for placing the comment on the table.

Dropping a table

Use the Drop Table panel to drop a table.

About this task

To drop a table:

Procedure

1. Select option DT on the Create/Drop/Label/Comment On Objects panel. The Drop Table panel is displayed, as shown in the following figure.

```
DB2 Admin --------------------- DB2X Drop Table ----------------------- 15:48
Command ===>

 DROP

 Schema. . . .          >         (default is D123)
 Name  . . . . TABLE07_TEST      > (? to look up)
```

Figure 101. The Drop Table panel (ADB26DT)

2. Specify the following values:
   - In the **Schema** field, specify the schema for the table that you want to drop. The schema-name must identify a schema that is described in the catalog.
   - In the **Name** field, enter a table name that you want to drop or enter a question mark (?) to look up existing table names using the Tables, Views, and Aliases panel.

3. Press Enter.
4. Follow the instructions on the Statement Execution Prompt panel (if enabled) to complete and run the SQL statement for dropping the table.

Using Drop Impact reports

When dropping DB2 objects, you can request Drop Impact reports to identify other DB2 objects, plans, and packages that are impacted by the action.

About this task

Drop Impact reports are useful tools that help you avoid dropping object that can adversely impact other DB2 objects, plans, and packages. Consider generating a Drop Impact report whenever you drop a DB2 object.

Tip: Use Drop Impact reports when you drop DB2 objects to avoid dropping objects that adversely impact other DB2 objects, plans, and packages.

Procedure

1. Issue the DROP line command on the Databases panel to drop a database, as shown in the following figure.

```
DB2 Admin ------------------- DSNB Databases ----------------- Row 1 to 3 of 3
Command ===>                                               Scroll ===> PAGE

Commands: GRANT  MIG  DIS  STA  STO  UTIL  CT
Line commands:
 T - Tables  S - Table spaces  X - Indexes  G - Storage group  ICS - IC status
 DIS - Display database  STA - Start database  STO - Stop database  A - Auth
 ? - Show all line commands
                         Storage  Buffer          Created       Index
Select Name      Owner   Group    Pool      DBID By        T E BPool     I
       *         *       *        *            * *         * * *         *
------ -------- -------- -------- -------- ------ -------- - - -------- -
       DBOC0001 NNAGAI   SYSDEFLT BP0        546 NNAGAI     U BP1       N
       DBOCA001 NNAGAI   SYSDEFLT BP0        545 NNAGAI     U BP1       N
DROP   DBOCMNN1 NNAGAI   SYSDEFLT BP0       1120 NNAGAI     E BP1       N
***************************** END OF DB2 DATA ******************************
```
Figure 102. Using the DROP command on the Databases panel (ADB21D)

2. Press Enter. The Drop Database panel (ADB26DD) is displayed, as shown in the following figure. If you set the default value for Drop Impact Reports to Yes, the field contains a YES value. If the **Display Drop Impact Report** is set to NO, change it to YES.

```
DB2 Admin --------------------- DSNB Drop Database --------------------- 15:38
 Command ===>

  DROP DATABASE

  Name     ===> DBOCMNN1 (? to look up)

  All objects in the database will be dropped.

  Display Drop Impact Report ===> YES   (Yes, No, or Batch)
```
Figure 103. The Drop Database panel (ADB26DD)

3. Press Enter to display the DROP Impact Analysis Summary panel (ADB2DIP). A portion of this panel is shown in the following figure.

```
DB2 Admin -------------- DSNB DROP Impact Analysis Summary -------------- 12:47
Command ===>                                                  Scroll ===> PAGE

SQL Statement: DROP DATABASE     "DBOCMNN1"
Line commands:  S - Show   blank - Suppress

   Items to                    Items to              Constraints to
   DROP or REVOKE    Count   Invalidate   Count    Remove                   Count
   --------------- -------   ---------- -------    -------------------- -------
S Databases . . . :      1 S Aliases . . :      1 S Check Constraints . :      0
S Table spaces  . :      3 S Packages  :        0 S Ref. Constraints . . :      0
S Tables  . . . . :      2 S Plans . . :        0 S Unique Constraints  :      4
S Aux. tables . . :      0              ======= S Masks . . . . . . . . :      0
S XML tables  . . :      0  Total  . . :        1 S Permissions . . . . . :      0
S History tables  :      0                                               =======
S Clone tables  . :      0                                   Total . . :      4
S Indexes . . . . :      4
S Authorizations  :      0
S Synonyms  . . . :      0
S Views . . . . . :      1
S Procedures  . . :      0
S Functions . . . :      0
S Triggers  . . . :      1
S User data types :      0
S Sequences . . . :      0
S Packages  . . . :      0
S Variables . . . :      0
                   =======
          Total . :     12
```

Figure 104. DROP Impact Analysis Summary panel (ADB2DIP)

4. Press Enter to display the DROP Impact Analysis Details panel (ADB2DIPD). A portion of this panel is shown in the following figure. This panel displays all objects that are impacted by dropping the object.

```
DB2 Admin ---------- DSNB DROP Impact Analysis Details ----- Row 1 to 17 of 17
Command ===>                                                  Scroll ===> PAGE

SQL Statement: DROP DATABASE     "DBOCMNN1"

Commands: RE-SORT   DROP
Line commands:  S - Show object   DRD - DROP RESTRICT on DROP

                                         Owner/
Sel Type   Object Name/Grantor>Grantee Schema     Note
    *      *                             *          *
--- ------ -------------------------- -------- ------------------------------
    D----- DBOCMNN1------------------- NNAGAI
    S        DBOCMNN1.TSOCM231          NNAGAI    UTS - PBG
     T         TBOCM231_TEACHER_PBR     NNAGAI
      ALI         NNGVAL                NNAGAI    Orphaned Alias
      UC          TEACHER_ID            NNAGAI    Primary key
      UC          TEACHER_ID1           NNAGAI    Unique key
      X           IU01_TEACHER_PBR      NNAGAI    Cluster
      X           IU02_TEACHER_PBR      NNAGAI
      J           MYTRIG2INT            NNAGAI
    S        DBOCMNN1.TSOCM232          NNAGAI    Segmented
     T         TBOCM232_TEACHER_PBG     NNAGAI
      UC          TEACHER_ID            NNAGAI    Primary key
      UC          TEACHER_ID1           NNAGAI    Unique key
      X           IU01_TEACHER_PBG      NNAGAI    Cluster
      X           IU02_TEACHER_PBG      NNAGAI
    S        DBOCMNN1.TSOCM233          NNAGAI    UTS - PBG
    V          NNVWTCH                  NNAGAI    View of a Table
```

Figure 105. Partial display of DROP Impact Analysis Details panel (ADB2DIPD)

On the DROP Impact Analysis Details panel, you can issue the following primary commands:

RE-SORT
Re-sort the table to its original sequence.

DROP
Proceed to drop the object.

Restriction: On the DROP Impact Analysis Details panel, you must type the Drop command on the primary command line and press Enter. You cannot issue the command by positioning the cursor on the DROP primary command and pressing Enter.

Sort
Sort the table based on using one or more columns.

On the DROP Impact Analysis Details panel, you can issue the following line commands:

S Show further details about an object.

DRD
Drop Restrict on Drop for the object.

Using Restrict on Drop
If a table has the Restrict on Drop attribute, users are restricted from dropping the object until the attribute is removed.

Occasionally, DB2 tables contain the Restrict on Drop attribute to prevent users from accidently dropping them. When attempting to drop one or more tables that have the Restrict on Drop attribute, DB2 Admin displays the Tables with Restrict on Drop panel, as shown in the following figure.

```
DB2 Admin -------------- DB2X Tables with Restrict on Drop -------- Row 1 of 1
Command ===>
Drop Table Restricted
SQL Statement: DROP DATABASE "VNDWLBD0"

DROP statement failed because one or more tables are defined with
RESTRICT ON DROP.

Commands:  DROP - DROP Restrict on Drop and DROP DATABASE
Line commands:  DRD - DROP RESTRICT on DROP

Sel Table Name         Owner    DB Name  TS Name  Note
    *                  *        *        *        *
--- ------------------ -------- -------- -------- ----------------
    EMP_PHOTO_RESUME   VNDWLB   VNDWLBD0 VNDWLBS3 Restrict on Drop
****************************** END OF DB2 DATA ******************************
```

Figure 106. Tables with Restrict on Drop panel (ADB26DDR)

DB2 Admin also displays this panel if a user attempts to drop a database or a table space that contains one or more tables that have the Restrict on Drop attribute.

To remove the Restrict on Drop attribute from a table, use the DRD line command. The DRD line command removes the Restrict on Drop attribute without dropping the table.

When dropping a database, table space, or table, you can use the DROP primary command to remove the Restrict on Drop attribute from the tables that are involved and then drop the database, table space, or table.

When the PROMPT option is used while dropping an object, the DROP statement for the object is displayed. You must select option 1 to run the DROP statement. If the DROP statement fails (with error code –672) because one or more tables have the Restrict on Drop attribute, the Tables with Restrict on Drop (ADB26DDR) panel is displayed. At this point, you can take one of the following actions:

- Use the DROP primary command to remove the Restrict on Drop attribute from the tables and run the DROP statement again. The DROP primary command generates an ALTER DROP RESTRICT ON DROP statement for each table, followed by a DROP statement.
- Use the DRD line command to remove the Restrict on Drop attribute for an individual table.
- Cancel and exit without running the DROP statement.

Creating a global variable

Use the Create Global Variable panel to create a new global variable.

Procedure

1. Select option CGV on the Create/Drop/Label/Comment On Objects panel. The Create Global Variable panel is displayed, as shown in the following figure.

```
ADBP6CGV  ---------------- DSNB Create Global Variable ----------------- 04:43
 Command ===> _____

 CREATE VARIABLE

 Schema  . . . . . ULVEMAN  >        (default is ULVEMAN)
 Name  . . . . . . TUJVCH128DUSER  > (? to look up)

 Data type . . . . VARCHAR           (Built-in type except: XML, ROWID, LOB)
 Data length . . . 128               (for CHAR, VARCHAR, GRAPHIC, VARGRAHIC,
                                      BINARY, or VARBINARY)
 Precision . . . . __               (1-53 FLOAT, 1-31 DECIMAL, or
                                     16 or 34 DECFLOAT)
 Scale . . . . . . __                (0-31 DECIMAL or 0-12 TIMESTAMP)

 FOR ? DATA  . . . _____           (BIT, SBCS, or MIXED)

 WITH TIME ZONE . ___                (Yes/No - for TIMESTAMP only)

 DEFAULT . . . . . USER                                              >
```

Figure 107. Create Global Variable panel (ADBP6CGV)

2. Specify the following values for the global variable:
   a. In the **Schema** field, enter the schema.
   b. In the **Name** field, enter the name.
   c. In the **Data type** field, enter the data type.

      Restriction: XML, ROWID, or LOB data types are not valid in this field.
   d. In the **Data length** field, enter the maximum length.
   e. In the **Precision** field, enter the precision. Precision only applies to FLOAT, DECIMAL, or DECFLOAT data types.
   f. In the **Scale** field, enter the scale. Scale only applies to DECIMAL, or TIMESTAMP data types.
   g. If applicable, in the **FOR ? DATA** field, enter the subtype for a CHARACTER data type.

    h. If applicable, in the **WITH TIME ZONE** field, enter the subtype for a
       TIMESTAMP data type.

    i. In the **Default** field, enter the default value.

3. Follow the instructions on the Statement Execution Prompt panel (if enabled) to
   complete and run the SQL statement for creating the new global variable.

Granting and revoking privileges on objects panel

Use the Grant or Revoke Privileges On Objects panel to issue GRANT and
REVOKE SQL statements.

About this task

The following example shows how to revoke privileges on a table.

To revoke privileges on a table:

Procedure

1. Select option 5 on the Execute SQL Statements panel. The Grant or Revoke
   Privileges On Objects panel is displayed, as shown in the following figure.

```
ADB2G min ----------- DB2A Grant/Revoke Privileges On Objects ---------- 13:2
Option ===>

                                                    DB2 System: DB2A
GRANT                                   REVOKE        DB2 SQL ID: SYSADM
  GG  - Storage group                     RG  - Storage group
  GD  - Database                          RD  - Database
  GS  - Table space                       RS  - Table space
  GT  - Table or view                     RT  - Table or view
  GC  - Column
  GP  - Plan                              RP  - Plan
  GL  - Collection                        RL  - Collection
  GK  - Package                           RK  - Package
  GZ  - System privilege                  RZ  - System privilege
  GR  - Buffer pool                       RR  - Buffer pool
  GH  - Schema                            RH  - Schema
  GE  - Distinct type                     RE  - Distinct type
  GF  - Function                          RF  - Function
  GO  - Stored procedure                  RO  - Stored procedure
  GJ  - JAR file                          RJ  - JAR file
  GQ  - Sequence                          RQ  - Sequence
  GGV - Global Variable                   RGV - Global variable

  CP  - Copy privileges
```

Figure 108. Grant or Revoke Privileges On Objects panel (ADB2G)

2. Specify RT in the **Option** field and press Enter. The Revoke Table Privileges
   panel is displayed, as shown in the following figure.

```
DB2 Admin ---------------- DB2X Revoke Table Privileges ---------------- 10:18
Command ===>

 REVOKE                                            DB2 SQL ID: ISTJE

 Enter any character in front of the privilege to revoke it from the user:

  ALL            INDEX          UPDATE
  ALTER          INSERT         REFERENCE
  DELETE         SELECT         TRIGGERS

 ON TABLE
  Owner  . . . MULTIPLE >
  Table  . . . ALL              >
FROM
  From . . . .          >
BY
  By . . . . . ISTJE

INCLUDING DEPENDENT PRIVILEGES
  Cascade revoke  . . . . . . . . . . YES (Yes/No)

  Report Revoke Impacts . . . . . . . YES (Yes/No)
  Report Dropped Synonyms & Aliases . . NO  (Yes/No)
```

Figure 109. Revoke Table Privileges panel (ADB2RT)

3. Specify the following information:
   - Type of privilege that you want to revoke
   - Owner name
   - Table name
   - User ID from which the privilege is being revoked (the **FROM** field)
   - User ID that is revoking the privilege (the **BY** field)

   When you issue a REVOKE command, you can choose to view a Revoke Impact Report. For example, on the Revoke Table Privileges panel in the previous figure, you can enter Y in the **Report Revoke Impacts** field. The report is displayed as a tree structure. The complete tree represents all of the authorizations or objects that will be lost or invalidated as a consequence of performing the REVOKE.

   Similarly, you can choose to view a Dropped Synonyms and Aliases Report by entering a Y in that field.

4. Press Enter to revoke the specified privilege.

Copying privileges from existing objects to other objects

Use the Copy privileges panel (ADBPCP) to copy privileges from existing objects to other objects.

About this task

When new objects are created, it is often necessary to grant privileges to the new objects, and often the same privileges from an existing object are needed for the new objects. The following example shows how to copy privileges from existing objects to other objects.

To copy privileges from existing objects to other objects:

Procedure

1. Enter the line command CP on the associated panel to copy privileges from the following object types:

> Aliases (ADB21A)
>
> Storage Groups (ADB21G)
>
> Databases (ADB21D)
>
> Table Spaces (ADB21S)
>
> Tables, Views, and Aliases (ADB21T).
>
> Schemas (ADB21H)
>
> Data Types (ADB21E)
>
> Functions (ADB21F)
>
> Stored Procedures (ADB21O)
>
> Sequence Objects (ADB21Q)
>
> Grant/Revoke Privileges On Objects (ADB2G)
>
> Version Scopes (ADB2C42)
>
> Global Variables (ADBP1GV)

2. Choose a method of copying privileges on the Copy Privileges panel (ADBPCP). There are three methods of copying privileges:

One-to-one
> All privileges from one source object are granted to one target object of the same kind. One-to-one results in GRANT statements for all privileges on one specific object to be built for granting authority to another specific object.

One-to-many
> All privileges from one source object are granted to multiple target objects of the same kind. One-to-many results in GRANT statements for all privileges on one specific object to be repeated for each of many other specific objects.

Many-to-many
> All privileges for each object in one set of source objects are granted to their counterpart objects in one set of target objects. Many-to-many results in GRANT statements for all privileges on a set of objects and their descendent objects to be built for granting authority to another set of objects.

> **Note:** The many-to-many method can produce GRANT statements to non-existent objects. When performed, these GRANTs produce SQLCODE -204, which is tolerated (because of the --#SET ACCEPT_RC statement which precedes these GRANT statements) and processing continues. You can leave these GRANTs in the DDL file (along with the --#SET statements) or remove them.

Restriction: If copy privileges are copied from source objects, for example, OBJECT1 to OBJECT2, OBJECT1 can have a maximum of 30000 GRANTS on it. More than 30000 GRANTS will not be processed as the stack allocated is 30000.

This table shows further detail on the three copying privileges methods:

Table 8. Three methods for copying privileges

	To One	To Many
From One	GRANTs from a single object are produced. Source object is provided on the panel. Target object is provided on the panel. No cascading the operation to dependent objects occurs.	GRANTs from a single object are produced. Source object is provided on the panel. Target objects are located by a version scope or quick-version scope. No cascading the operation to dependent objects occurs.
From Many		GRANTs from multiple objects and their dependent objects are produced. Source objects are located by a version scope or quick-version scope. Target objects are determined by masking the source object names. GRANTs to certain object types can be excluded.

```
ADBPCP ---------------------- DB2X Copy Privileges -------------------- 16:08
Option ===>


 1  One-to-one   - Copy from one object to another
 2  One-to-many  - Copy from one object to many others
 3  Many-to-many - Copy from many objects to many objects

From one object specification:
  Schema/Qual  . . . . J148286  >
  Name . . . . . . . . ALAD7G02            >  (? to look up)
  Type . . . . . . . . AL                 (SG,DB,TS,TB,VW,AL,DT,FU,SC,SP,SQ,GV)

To one object specification:
  Schema/Qual  . . . . J148286  >
  Name . . . . . . . . ONAVIEW             >  (? to look up)

Many objects specification:                (A version scope or as a quick scope)
  Owner  . . . . . . . J148286  >
  Name . . . . . . . . *                   >  (? to look up)
  Quick scope type . . AL                 (SG,DB,TS,TB,VW,AL,DT,FU,SC,SP,SQ,GV)

Options:
  Run SQLID  . . . . . . . .
  Generate online  . . . . . NO      (Yes/No)
  As work statement list . . NO      (Yes/No)
  Data set name  . . . . . . SOURCE.DDL                                   >
    Data set disposition . . OLD      (OLD, SHR, or MOD)
  Prompt to run SQL  . . . . YES     (Yes/No. For online mode only)

GRANT options for Many-to-many:
  GRANT use OF STORAGE GROUP . . Y (Y,N,A,R)
  GRANT access ON DATABASE . . . N (Y,N,A,R)
  GRANT access ON TABLESPACE . . A (Y,N,A,R)
  GRANT access ON TABLE  . . . . R (Y,N,A,R)
  GRANT access ON VIEW . . . . . Y (Y,N,A,R)
  GRANT access ON SCHEMA . . . . N (Y,N,A,R)
  GRANT USE OF DISTINCT TYPE . . A (Y,N,A,R)
  GRANT access ON FUNCTION . . . R (Y,N,A,R)
  GRANT access ON PROCEDURE  . . Y (Y,N,A,R)
  GRANT access ON SEQUENCE . . . N (Y,N,A,R)
  GRANT access ON VARIABLE . . . Y (Y,N,A,R)
 BP - Change batch job parameters
```

Figure 110. Copy Privileges panel (ADBPCP)

Revoking system authority from an SQLID

Use the System Privileges Authorization panel to revoke system authority from an SQLID.

About this task

The following example shows how to revoke system authority from an SQLID and run a Revoke Impact Report.

To revoke system authority from an SQLID and run a report:

Procedure

1. On panel ADB21, System Catalog, enter the authid you want to revoke in the Grantee field with and then issue the **AO - Authorization options** command.
2. When the authorization options are displayed on panel ADB21, System Catalog, issue the **UA - User authorizations** command. A summary displays for the SQLID on panel ADB2AUS,User Authorizations Summary

3. From panel ADB2AUS, issue the **AU** line command.

```
ADB2AUS n ----------- DB2X User Authorizations Summary ----- Row 1 to
  Command ===>                                         Scroll
  Authorities held by C222333%
  Authority includes SYSADM
  Commands:  AU AP ALL AE AI
  Line commands:  AU - User Only  AP - All PUBLIC  ALL - All Authorizat
                  AE - Explicit to User  AI - Implicit to User
  Sel Type              Explicit    Implicit     PUBLIC      Total


--- ---------------- ----------- ----------- ----------- -----------

  AU  System                 2           0           1           3
      Storage group          0          21          15          36
      Database               0         306          57         363
      Table space            1           0         105         106
      Table                  1         305        2768        3074
      Column                 0           3           0           3
      Plan                   4          47         220         271
      Collection             0           0           2           2
      Package               44         459         218         721
      Function               0           4           1           5
      Buffer pool            0           0           8           8
      Data type              0           0           1           1
      JAR                    0           0           0           0
      Stored procedure       0           4          41          45
      Schema                 0           0           2           2
      Sequence               0           1           0           1

****************************** END OF DB2 DATA *********************
```

Figure 111. User Authorizations Summary panel (ADB2AUS)

4. Start the REVOKE process and its associated Revoke Impact Report by issuing the **R - Revoke** line command from panel ADB2AZ, System Privileges Authorizations.

```
ADB2AZ in ------- DBAA System Privileges Authorizations on objects -- Row 1 to 5 of 5
Command ===>                                             Scroll ===> CSR

Commands: REVOKE  GRANT  SYSAUTH
Line commands:
 R - Revoke  GR - Grant           B B  CREATE   : S B M M D E S S S S S D A
 I - Interpretation               I S          S T I O O E X Q Y Y Y Y A C
 RE - Grantee role                N D     A T E O N N N B P L S S S D T C
 RR - Grantor role                D S     L M C S D 1 2 U L A A C O B A E
                                   A   D D   I T U P A     G A D D T P A A S
                      G            H D   B B S A A R A G   S I M M R R D C S
Sel Grantor  Grantee  T Grant date G D   A C G S B E C T   E N     L   M C C
    *        *        * *          * * * * * * * * * * * * * * * * * * * * *
--- -------- -------- - ---------- - - - - - - - - - - - - - - - - - - - - -
R   BISVT    SUNDARI    2008-02-13 S                       Y           Y   Y
    BISVT    JSTEWART   2008-08-21 S                       Y G
    BISVT    PATSHIM    2008-09-15 S                       Y   G
    BISVT    STEWART    2009-01-28 S                       Y Y         Y Y
    BISVT    PHOENIX    2009-03-13 S                       Y
****************************** END OF DB2 DATA ******************************
```

Figure 112. System Privileges Authorizations panel (ADB2AZ)

5. Enter YES in the **Report Revoke Impacts** field on panel ADB2RZ, Revoke System Privileges.

```
AADB2RZ in --------------- DB2X Revoke System Privileges --------------- 07:05
Command ===> _____

 REVOKE                                              DB2 SQL ID: SMITHJ

 Enter any character in front of the privilege to revoke it from the user:

 _ SYSADM        _ BSDS          _ CREATESG      _ STOPALL
 _ SYSOPR        _ CREATEDBA     _ DISPLAY       _ STOSPACE
 _ BINDADD       _ CREATEDBC     _ RECOVER       _ TRACE
 _ MONITOR1      _ MONITOR2      _ CREATEALIAS   _ SYSCTRL
 _ BINDAGENT     _ ARCHIVE       _ CREATETMTAB   _ DEBUGSESSION
 _ EXPLAIN       _ SQLADM        _ DBADM         _ DATAACCESS
 Y ACCESSCTRL    _ CREATE_SECURE_OBJECT

 FROM
   From . . . . . . . ACCESSCTRL                                        >
 BY
   By . . . . . . . . _____     >
 INCLUDING DEPENDENT PRIVILEGES
   Cascade revoke . . ___ (Yes/No)

 Report Revoke Impacts . . . NO   (Yes/No)
 Report Dropped Synonyms & Aliases . . NO   (Yes/No)
```

Figure 113. Revoke System Privileges panel (ADB2RZ)

6. Check the details on panel ADB2RIP, Revoke Impact Report.

```
ADB2RIP n -------------- DB2X Revoke Impact Report --------------- Row 1 of 1
 Command ===> _____ Scroll ===> PAGE


 Line commands:  I - Interpretation
                              Owner/
 S     Grantee  G Resource N/  O Schema/  Grantor/ G H Privileges/
   Lvl          T Collection  T P/K Name Binder  T G Effect
 - --- -------- - ----------- - -------- -------- - - ------------------------
 * 0   PACKADM                Z           VNDRG    S      Y
 ****************************** END OF DB2 DATA ******************************
```

Figure 114. Revoke Impact Report panel (ADB2RIP)

Chapter 10. Reconstructing SQL (DB2 Admin Reverse Engineering)

DB2 Admin can generate the SQL statements that are required to re-create a DB2 object.

This process, which is called *reverse engineering*, is accomplished by using the GEN command to extract the SQL for an object from the DB2 catalog.

Extracting the DDL for an object before changes are made is helpful for ensuring that the changes are applied to the current definition. The DDL also ensures that the original object definitions are available for fallback purposes.

When extracting objects from databases, table spaces, and tables, you can also generate all dependent objects, including: table spaces, tables, indexes, views, synonyms, aliases, referential constraints, table check constraints, and triggers. When extracting objects from schemas, you can extract the associated distinct types, sequences, functions, global variables, and stored procedures. Alternatively, you can specify objects that you want to exclude from the generated DDL as well.

In addition to extracting the DDL for objects, you can also generate the DCL for all authorizations to the objects and the DML for the catalog statistics for the objects.

You can generate the SQL statements online or with a batch job. Batch jobs are recommended when you extract many objects from a large catalog.

Topics:

Generating SQL to re-create a DB2 object

About this task

To generate SQL to re-create DB2 objects:

Procedure
1. From the DB2 Admin Main Menu, specify option 1. The System Catalog panel is displayed.
2. Select one of the following options that supports the ability to specify the GEN command to reverse engineer objects.
   - Databases (option D)
   - Table spaces (option S)
   - Tables, views, and aliases (option T)
   - Aliases (option A)
   - Synonyms (option Y)
   - Schemas (option H)
   - User defined data types (option E)
   - Functions (option F)

- Storage groups (option G)
- Stored procedures (option O)
- Triggers (option J)
- Sequences (option Q)
- Global variables (option GV)

Tip: You can use the fields at the bottom of the panel to specify search criteria to filter or limit the number of objects that are displayed.

3. Generate SQL.
   - To generate the SQL for a single object that is listed, issue the GEN line command for the object.
   - To generate the SQL for all of the listed objects, issue the GEN primary command.

   The following figure shows the GEN line command being issued against a database.

```
DB2 Admin ------------------- DB2X Databases ---------------------- Row 1 of 4
Command ===>                                              Scroll ===> PAGE

Commands: GRANT  MIG  DIS  STA  STO  UTIL
Line commands:
 T - Tables  S - Table spaces  X - Indexes  G - Storage group  ICS - IC status
 DIS - Display database  STA - Start database  STO - Stop database  A - Auth
 ? - Show all line commands
                          Storage  Buffer         Created       Index
Select Name      Owner    Group    Pool      DBID By       T E BPool    I
       *         *        *        *           * *        * * *         *
------ -------- -------- -------- -------- ------ -------- - - -------- -
GEN    DSN8D81A DSCGDB2  DSN8G810 BP0        258 ISTJE      E BP2       Y
       DSN8D81E DSCGDB2  DSN8G810 BP1        260 ISTJE      U BP2       Y
       DSN8D81P DSCGDB2  DSN8G810 BP0        259 ISTJE      E BP2       N
       DSN8D81U DSCGDB2  DSN8G81U BP1        261 ISTJE      E BP2       N
****************************** END OF DB2 DATA ******************************
```

Figure 115. Databases panel (ADB21D) - Example of issuing the GEN command to reverse engineer objects

Tip: The DDL line command is a convenient alternative to using the GEN command when you want to view only the DDL for a single object in the DB2 catalog. The DDL command does not provide the additional options that the GEN command does for extracting additional information, such as constraints, authorizations, or dependent objects like triggers, labels, or comments. The DDL line command is valid anywhere that the GEN line command is valid with these exceptions:

- It is not valid on the Schemas panel (Option 1.H).
- It is valid on the Indexes panel (Option 1.X).

Note: When a native SQL procedure statement size is near the 2 MB boundary, sometimes GEN cannot generate the native SQL procedure statement DDL. Two scenarios can occur when GEN might not be able to generate the native SQL procedure DDL:

- The native SQL procedure statement is created by GEN by first constructing the native SQL procedure options from the catalog fields (other than sysroutines.text) and appending the native SQL procedure SQL-routine-body that is stored in sysroutines.text. Sometimes the resulting DDL statement exceeds 2 MB. This might occur because more options are generated by GEN (such as keep default option values, when the "DB2 defaults handling" option is set to Keep) than were specified when the native SQL procedure

was created. When the 2 MB is exceeded in this scenario, GEN will issue the ADB1915W warning message and generate the native SQL procedure DDL as it is stored in DB2. The resulting DDL for the native SQL procedure object is the exact contents of the sysroutines.text field. If masking or an override was specified (such as change owner, change schema, or Run SQLID), the ADB1916E error message will be issued instead and GEN processing will stop. This is because GEN cannot complete the request within 2 MB for the native SQL procedure DDL with the specified masks or overrides.

- GEN attempts to format each DDL statement so it is easy to read. Sometimes during the formatting process the extra bytes added for formatting cause the formatted statement length to exceed 2 MB. When this occurs, GEN will issue the ADB1919W warning message and generate unformatted DDL for the native SQL procedure. If masking or an override was specified (such as change owner, change schema, or Run SQLID), the ADB1920E error message will be issued instead and GEN processing will stop. This is because GEN cannot complete the request within 2 MB for the native SQL procedure DDL with the specified masks or overrides.

The generated statement terminator was ? (question mark) for releases earlier than DB2 Admin Version 11.1 and is the ` (grave accent) for DB2 Admin Version 11.1 and later releases.When you use either the GEN line command or the GEN primary command, the Generate SQL from DB2 catalog panel is displayed, as shown in the following figure.

```
ADB2GEN n -------------- DB2X Generate SQL from DB2 catalog ------------- 11:34
Option ===>

Generate SQL statements for database DBFSSGEN          DB2 System: DB2X
                                                       DB2 SQL ID: JSMITH

SQL statement types to be generated from the DB2 catalog:
   CREATE DATABASE  . . . . Y (Y,N)   GRANT access ON DATABASE . . Y (Y,N,A,R)
   CREATE TABLESPACE . . . Y (Y,N)   GRANT access ON TABLESPACE . Y (Y,N,A,R)
   CREATE TABLE . . . . . . Y (Y,N)   GRANT access ON TABLE  . . . Y (Y,N,A,R)
   CREATE VIEW  . . . . . Y (Y,N,D) GRANT access ON VIEW . . . . Y (Y,N,A,R)
   CREATE INDEX . . . . . . Y (Y,N)   ALTER TABLE ADD FOREIGN KEY. Y (Y,N,D)
   CREATE SYNONYM . . . . . Y (Y,N)   LABEL ON . . . . . . . . . . Y (Y,N)
   CREATE ALIAS . . . . . . Y (Y,N)   COMMENT ON . . . . . . . . . Y (Y,N)
   CREATE TRIGGER . . . . . Y (Y,N,D) REBIND PLAN/PACKAGE  . . . . Y (Y,N,D)
   CREATE MASK  . . . . . . Y (Y,N)   ALTER TABLE ACTIVATE CONTROL Y (Y,N)
   CREATE PERMISSION  . . . Y (Y,N)
   CREATE STORAGE GROUP . . Y (Y,N)   GRANT use OF STORAGE GROUP . Y (Y,N,A,R)

 New names/values for generated SQL: (leave blank to use current values)
   Object schema  . . . . .             > Run SQLID  . . . . . . . .
   Object grantor . . . . .             >
   Alloc TS size as . . . . DEFINED     (DEFINED, USED, or ALLOC)
   Database name  . . . . .
   Storage group for TS . .             > Storage group for IX . . .           >
   Target DB2 version . . .             (Current DB2 version: 1115)
   Use Masking  . . . . . . NO          (Yes/No)
   Use Exclude Spec . . . . NO          (Yes/No)
   Target cat qualifier . .             > (Default is SYSIBM)
   Generate catalog stats . NO          (Yes,No,Only)
     Statistics tables  . . ALL         (All or Select. Default is All)
   Include DB2 pending chgs NO          (Yes,No,Alter,Only)
   PBG NUMPARTS value . . . EXISTING    (Defined, Existing)
   PBG LOB objects  . . . . COMPUTED    (Computed, Implicit)
   Generate index cleanup .             (Yes,No,Only)

 SQL output data set and execution mode:
   Add to a WSL . . . . . . NO          (Yes/No)
   Data set name  . . . . .
     Data set disposition . OLD         (OLD, SHR, or MOD)
   Execution mode . . . . . BATCH       (BATCH or TSO)
   Commit statements per  .             (Db, tS, Tb, All, None. Default is All)
   DB2 defaults handling  .             (Keep, or Remove. Default is Keep)
   Prompt to run SQL  . . . NO          (Yes/No. For TSO mode and no WSL)
   Include SQL comments . . NO          (Yes/No. For BATCH mode and no WSL)

 DB2 Command output data set:
   Data set name  . . . . .
     Data set disposition . OLD         (OLD, SHR, or MOD)

 BP - Change batch job parameters
  G - Change additional parameters
```

Figure 116. Generate SQL from DB2 Catalog panel (ADB2GEN)

4. Fill in the fields in the Generate SQL from the DB2 catalog panel, as shown in
   the previous figure. In most cases, the valid values are Y and N. For detailed
   descriptions of the fields, refer to the online help for the panel. For DB2 9 NFM
   or later, the values available for the GRANT access statement types and
   GRANT use OF STORAGE GROUP are:

   Y Generate GRANT statements for authorizations and roles

   N Do not generate any GRANT statements

   A Generate GRANT statements for authorizations

   R Generate GRANT statements for roles

   The fields are grouped:

- In the first set of fields, specify whether a CREATE statement is to be generated for the requested objects and dependent objects of the requested objects, where applicable:

CREATE DATABASE

> A value of Y specifies that CREATE statements for all of the explicitly requested databases are to be generated.

> When you also request to generate storage groups, statements are generated for the default storage group.

CREATE TABLESPACE

> A value of Y specifies that CREATE statements for all of the table spaces that are identified during processing are to be generated, which includes both explicitly and implicitly requested table spaces. For example, if you specify the GEN command for a database and specify Y in the **CREATE TABLESPACE** field, a CREATE statement will be generated for each table space that resides in the database.

CREATE TABLE

> A value of Y specifies that CREATE statements for all of the tables that are identified during processing are to be generated, which includes both explicitly and implicitly requested tables.

CREATE VIEW

> A value of Y specifies that CREATE statements for all of the views that are identified during processing are to be generated, which includes both explicitly and implicitly requested views.

> Specify D to extract views without DB2 Admin checking whether all other objects used in the view are also being generated. This option significantly reduces the resource consumption when running on large DB2 catalogs.

CREATE INDEX

> A value of Y specifies that CREATE statements for all of the indexes that are identified during processing are to be generated, which includes both explicitly and implicitly requested indexes.

CREATE SYNONYM

> A value of Y specifies that CREATE statements for all of the synonyms that are identified during processing are to be generated, which includes both explicitly and implicitly requested synonyms.

CREATE ALIAS

> A value of Y specifies that CREATE statements for all of the aliases that are identified during processing are to be generated, which includes both explicitly and implicitly requested aliases.

CREATE TRIGGER

> A value of Y specifies that CREATE statements for all of the triggers that are identified during processing are to be generated, which includes both explicitly and implicitly requested triggers.

CREATE MASK

> A value of Y specifies that CREATE statements for all of the masks that are identified during processing are to be generated, which includes both explicit and implicit masks.

CREATE PERMISSION

> A value of Y specifies that CREATE statements for all of the

permissions that are identified during processing are to be generated, which includes both explicit and implicit permissions.

CREATE STORAGE GROUP
A value of Y specifies that CREATE statements for all of the storage groups that are identified during processing are to be generated, which includes both explicit and implicit storage groups.

GRANT access ON DATABASE
Generates a GRANT access ON DATABASE statement in the SQL.

GRANT access ON TABLESPACE
Generates a GRANT access ON TABLESPACE statement in the SQL.

GRANT access ON TABLE
Generates a GRANT access ON TABLE statement in the SQL.

GRANT access ON VIEW
Generates a GRANT access ON VIEW statement in the SQL.

ALTER TABLE ADD FOREIGN KEY
Specify D to extract FOREIGN KEYS for tables that are dependent on the tables being extracted.

LABEL ON
Generates a LABEL ON statement in the SQL.

COMMENT ON
Generates a COMMENT ON statement in the SQL.

REBIND PLAN/PACKAGE
Generates REBIND commands for plans and packages. These REBIND commands are written to the data set that is specified in the **DB2 Command output file: Data set name** field.

ALTER TABLE ACTIVATE CONTROL
Activates an enabled masked column. A column mask can be created as enabled or disabled for column access control. An enabled column mask does not take effect until the ALTER TABLE statement with the ACTIVATE COLUMN ACCESS CONTROL clause is used to activate column access control for the table.

GRANT use OF STORAGE GROUP
Generates a GRANT USE OF STOGROUP statement in the SQL.

- In the second set of fields, specify the new names or values to be used in the generated SQL:

Object schema
Specify a new object schema. If specified, the new schema is used whenever an object is created.

Run SQLID
Specify the SQL ID to be used when creating objects. The SQL ID that is specified must have the privileges that are necessary to create objects, such as an administrative type of SQL ID that has been defined. If you specify a value of <NONE>, no SET CURRENT SQLID statements are generated in the DDL. If you leave the field blank, a SET CURRENT SQLID statement is generated in the DDL before each object that is created (where possible, the SQL ID that was originally used to create the object is used).

If you specify an SQLID of <NONE>, the following is true if you use synonyms:

- If the creator of the synonym is the same as the creator of the table on which the synonym is defined, an executable CREATE SYNONYM statement is generated.
- If the creator of the synonym is not the same as the creator of the table on which the synonym is defined, the SQLID that created the SYNONYM is extracted from the catalog and both the SET SQLID and CREATE SYNONYM statements are created, but commented out. An informational message is issued. Be aware that other generated statements might fail due to these statements being commented out (for example, a view that is defined using the synonym).

The other DB2 Admin functions where you can specify a RUN SQLID value include the Rename Database, ALT, Migrate, and Change Management functions.

Object grantor
The grantor of the object.

Alloc TS size as
Specifies how to generate the primary quantity. The following values are valid:

DEFINED
Uses the size defined in the catalog.

USED
Uses the size that is actually used. This option requires you to run the STOSPACE utility for the storage groups for the objects being generated.

ALLOC
Uses the allocated size. This option requires you to run the STOSPACE utility for the storage groups for the objects being generated.

Database name
Specify a new database name for the objects (except when initiated using a primary command from a list of databases).

Storage group for TS
Specify a new storage group for the table spaces.

Storage group for IX
Specify a new storage group for the indexes.

Target DB2 version
Specify the DB2 level for the generated SQL statements, if different from the current DB2 level. The DB2 level format is vvrm, where vv=version, r=release, and m=modification level. The current DB2 level is the default.

Important: Sometimes SQL syntax support is removed from DB2. Specifying the correct target DB2 version ensures that the generated SQL will be valid for the target DB2 subsystem. For example, for CREATE INDEX, DB2 11 supports EXCLUDE NULL KEYS, but DB2 10 does not.

The following values are examples of valid level values:

1010
DB2 10 compatibility mode (CM8)

1012
> DB2 10 compatibility mode (CM9)

1013
> DB2 10 enabling NFM

1015
> DB2 10 NFM

1110
> DB2 11 compatibility mode (CM)

1113
> DB2 11 enabling NFM

1115
> DB2 11 NFM

Example: Suppose that your current DB2 level is DB2 11 NFM, but you want to generate SQL that runs on a DB2 10 NFM system. Set 1015 as the target DB2 version.

Note: The IN DD run parameter DB2REL uses the same format and values as the Target DB2 version option. When DB2 Admin generates a GEN batch job, it picks up the DB2 release level from an SQL CONNECT statement and uses that release level value in the generated job. It is recommended that you use the generated job as the base for defining customized GEN jobs.

Include DB2 pending chgs
> Specify additional methods of including DB2 pending changes. The valid values are:

Yes
> (default) Include the DB2 pending changes when generating CREATE statements for table spaces and indexes.

No Generate SQL comments that contain ALTER statements for the DB2 pending changes. The DB2 pending changes are not included when generating CREATE statements for table spaces and indexes.

Alter
> Generate ALTER statements for the DB2 pending changes.

Only
> Only generate ALTER statements for the DB2 pending changes. No other SQL (such as CREATE statements) will be generated.

PBG NUMPARTS value
> The value for the NUMPARTS clause of a partition-by-growth (PBG) table space when the table space is recreated. Valid values are the following:

Defined
> The NUMPARTS clause will be generated with the value that was used when the table space was created.

Existing
> The NUMPARTS clause will be generated with the value that currently exists. The existing value includes any added partitions. This value can be different from the value that was defined when the table space is created This is the default.

PBG LOB objects
Specify whether the auxiliary objects for LOB columns in a partition-by-growth (PBG) table space are to be recreated implicitly or explicitly. Valid values are the following:

Computed
The auxiliary objects will be created explicitly if all of the DB2-required auxiliary objects exist and were created explicitly. This is the default.

Implicit
The auxiliary objects will be created implicitly by DB2.

Generate index cleanup
Specify index cleanup options. The valid values are:

Yes
Generate DML statements for the DB2 SYSINDEXCLEANUP table.

No Do not generate DML statements for the DB2 SYSINDEXCLEANUP table. This is the default value.

Only
Generate DML statements only for the DB2 SYSINDEXCLEANUP table. No other DDL (such as CREATE statements) or DML (such as catalog statistics) will be generated.

Use Masking
Specify Y to enable masking.

Use Exclude Spec
Specify Y to select or edit an existing exclude specification. The specification enables you to select objects to exclude from the generated DDL.

Target cat qualifier
Specify the qualifier to be used in the INSERT, UPDATE, and DELETE statements for updating catalog statistics and for index cleanup settings.

Generate catalog stats
Specify whether to generate catalog statistics, which causes INSERT, UPDATE, and DELETE statements that modify the catalog statistics to be included in the DDL file. The valid value are:

Y Generate DDL and catalog statistics.

N Generate DDL only. Do not generate catalog statistics.

O Generate catalog statistics only. Do not generate DDL.

The statistic fields that are generated are those that are associated with the objects that are being generated. (The complete list of statistics fields are those fields that are set by RUNSTATS that can be modified and the five statistics columns for table functions in SYSROUTINES, which are not set by RUNSTATS.)

Statistics tables
Specify All (Default) or Select to specify which statistics to generate. If you specify "Select," you can choose catalog tables from the

Catalog Statistics Tables panel (ADBPGEN2) that appears, then the SQL DML statements that are generated are for only the DB2 catalog tables that you selected.

- In the third set of fields, specify the output file and execution mode options:

Add to work stmt list
: Specify Y to save the output to a work statement list data set. Specify N to suppress work statement list output.

Data set name
: Specify the data set in which DB2 Admin should place the generated SQL. It must be an valid SPUFI input data set name or SYSOUT=x. The default is SYSOUT=*. If you leave the field blank, the command output is created as comments in the output file.

Data set disposition
: Specify the disposition of the output data set.

Execution mode

> **BATCH**
> : Specify BATCH to run it as a batch job. If you specify an execution mode of BATCH, DB2 Admin generates a batch job and displays the job in an ISPF edit session, ready for any modifications that you need to make before submitting the job for execution.

> **TSO**
> : Specify TSO to run the SQL generation online. If you specify TSO, DB2 Admin generates the SQL statements online and displays the results.

Commit statements per
: Specify how often an SQL COMMIT statement is added to the generated SQL. Valid values are:

> **D** Commit statements are run for each database.
>
> **S** Commit statements are run for each table space.
>
> **T** Commit statements are run for each table.
>
> **A** Commit statements are run for all objects (default).
>
> **N** Commit statements are never run.

DB2 defaults handling
: Specify whether DB2 default parameters should be removed or kept in the generated SQL. Valid values are:

> **K** Keeps DB2 default parameters (default).
>
> **R** Removes DB2 default parameters.

Prompt to run SQL
: Specifies that after the SQL edit session, a prompt displays that allows you to choose whether to run the SQL immediately. This option only applies when you are using TSO mode without WSL. Valid values are:

> **Y** After the SQL edit session, display a prompt that allows you to choose whether to run the SQL immediately.
>
> The maximum number of SQL statements that are allowed is 8120. The maximum length of an SQL statement is 2097152 bytes (2 MB).

> > **N** Do not display a prompt after the SQL edit session (default).

- In the last set of fields, specify the following options for the command output file:

  Data set name
  > Specify the data set in which DB2 Admin should place the generated REBIND commands if REBIND PLAN/PACKAGE is selected.

  Data set disposition
  > The disposition of the output data set.

  Restriction:
  - DB2 Admin does not extract IDCAMS DEFINE CLUSTER statements for VCAT-defined table spaces and indexes.
  - When you reconstruct a stored procedure that is implemented in SQL, DB2 Admin cannot recover the original procedure body and replaces the original procedure body with the string "LEAVE L0". The procedure body cannot be recovered because it is not stored in the catalog. This occurs only for a non-native SQL procedure stored procedure that is implemented in SQL (SQL - external).

  Tip: The ability to generate actual allocated space or actual used space allocations depends on information in the DB2 catalog. The actual data set sizes for table spaces and index spaces are not retrieved. Set the **Alloc TS size as** field to ALLOC or USED only if you have recently run STOSPACE and RUNSTATS for the selected objects.

Using parameters in generated SQL

In some cases, you might need to specify special parameters to enable the GEN function.

IMPLQUALMETHOD
> The **IMPLQUALMETHOD** parameter enables the GEN function to generate the CURRENT SQLID for views created prior to DB2 V9 and for views with unqualified synonyms or aliases. Issue the G primary command on the ADB2GEN panel to display the Change Additional Generate Parameters panel. The value you specify for the View CURRENT SQLID method field will be used to set the **IMPLQUALMETHOD** option in the GEN batch job.

> **Values:**

> > **O** The GEN function searches the DB2 catalog for objects with the unqualified name. If multiple objects are found, the GEN function will use the qualifier of the dependent table for the generated SET CURRENT SQLID statement.

> > **C** The GEN function searches the DB2 catalog for objects with the unqualified name. If multiple objects are found, the GEN function will use the qualifier of the view for the generated SET CURRENT SQLID statement.

Generating SQL using wildcard characters

When you reverse engineer objects and have the SQL statements generated in batch mode, you can use wildcard characters in the qualifiers and names of the objects to be extracted, which gives you the ability to have the DDL extracted based on strings that have a certain pattern.

The GEN operation supports the use of request parameters that name the specific objects that are to be generated. The request for an object is specified by providing values for three keywords: TYPE, QUAL, and NAME. For example, the following request generates the DDL for database DSNDB04 and all of the objects that it contains:

```
TYPE='DB',QUAL='',NAME='DSNDB04';
```

The VERSION attribute is only for an native SQL procedure and indicates which native SQL procedure version or versions to extract. The VERSION attribute can be used to specify a specific version to extract, to extract the active version, or all versions.

```
TYPE='SP',QUAL='DEMBIN2',NAME='MYSTP',VERSION='V1';
```

Note: VERSION='*' will extract all versions. QUAL='TEST',NAME='*' will extract all active stored procedures within schema TEST. If the version is omitted, or is set to blank, the active version will be extracted.

The values for the qualifier and name can contain zero or more of the following wildcard characters:
- Minus sign (-) represents any single character.
- Percent sign (%) or asterisk (*) represents one or more characters.
- Any other character represents a single occurrence of itself.

The rules for the wildcard characters follow the rules that are used for the LIKE predicate.

The following table shows the values to specify in the TYPE, QUAL, and NAME keywords for each type of object:

Table 9. The keyword values of the request parameters for each object type

Object Type	TYPE	QUAL	NAME	Notes
Database	DB	n/a	*dbanme*	
Table space	TS	*dbname*	*tsname*	
Table	TB	*creator*	*tbname*	
Global Variable	GV	*schema*	*gvname*	For DB2 Version 11 or later
View	VW	*creator*	*vwname*	
Alias	AL	*creator*	*aliasname*	
Index	IX	*creator*	*ixname*	
User-defined data type	DT	*schema*	*udtname*	
User-defined function	FU	*schema*	*udfname*	
Stored procedure	SP	*schema*	*stpname*	
Sequence	SQ	*schema*	*seqname*	
Schema	SC	*schema*	n/a	
Trigger	TG	*schema*	*tgname*	
Storage group	SG	n/a	*sgname*	
Synonym	SY	*creator*	*syname*	
Trusted context	TC	n/a	*tcname*	
Role	RO	n/a	*roname*	

The request parameters are specified in a data set with a DD name of IN. The request parameters must follow the run parameters in the data set.

Restriction: Modifying the run parameters in the IN file is not supported.

Modify the JCL that is generated to reverse engineer objects or modify the JCL that is provided in sample program ADBGEN to specify names with wildcard characters. The following figure shows an example of the sample program. Note that the semicolon (;) after the tgtdb2 parameter in the example ends the list of run parameters. What follows that are request parameters.

```
//GENSQL   EXEC PGM=IKJEFT01,DYNAMNBR=100
//STEPLIB  DD DISP=SHR,DSN=ADBB10.ISPLLIB
//         DD DISP=SHR,DSN=DSN.DSNA.SDSNEXIT
//         DD DISP=SHR,DSN=DSN.DSNA.SDSNLOAD
//         DD DISP=SHR,DSN=AUTHSW.ISPLLIB
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
  DSN SYSTEM(DSNA)
  RUN PROG(ADB2GEN) PLAN(ADB) PARMS('/REBIND')
  END
/*
//SYSPRINT DD SYSOUT=*
//SQL      DD SYSOUT=*,DCB=(RECFM=FB,LRECL=80)
//IN       DD *
  DB2SYS   = 'DSNA',
  DB2ALOC  = '',
  DB2SERV  = 'DSNA',
  DB2AUTH  = 'SINNOTT',
  DB2REL   = '1013',
  GENSG    = 'Y',
  GENDB    = 'Y',
  GENTS    = 'Y',
  GENTABLE = 'Y',
  GENVIEW  = 'Y',
.
.
.
  NEWGRANTOR = '',
  SPCALLOC = 'DEFINED',
  TGTDB2   = '';
  TYPE='DB',QUAL='',NAME='DSNDB04';
```

Figure 117. Sample program ADB2GEN to generate SQL in batch

Sample output from generating SQL

The figure in this topic is an example of the SQL that is generated when you use the GEN command to reverse engineer objects.

```
------------------------------------------------------------------------
--                                                                    --
-- Database 2 Administration Tool (DB2 Admin) , program 5655-W34      --
--                                                                    --
-- ADB2GEN  -  Extract object definitions from the DB2 Catalog tables --
--                                                                    --
-- Input prepared on : DSNA (1015 )   Extract time : 2013-05-14 07:37 --
--                                                                    --
-- Catalog values overridden : none
--                                                                    --
-- Generate : SG=N DB=Y TS=Y TB=Y VW=Y IX=Y SY=Y AL=Y LB=N CM=N FK=N  --
--            TG=Y UT=N UF=N SP=N SQ=N RO=N TC=N MK=Y PM=Y AC=Y        --
-- Grants   : SG=N DB=N TS=N TB=N VW=N SC=N UT=N UF=N SP=N SQ=N        --
--                                                                    --
------------------------------------------------------------------------
------------------------------------------------------------------------
--                                                                    --
-- ADB2GEN: Generate DDL for Database DSNDB06                         --
--                                                                    --
------------------------------------------------------------------------
--
--
------------------------------------------------------------------------
------------------------------------------------------------------------
--
  SET CURRENT SQLID='SYSIBM';
--
--#SET ACCEPT_RC 0 -618
--
  CREATE DATABASE DSNDB06
    INDEXBP     BP0
    CCSID       EBCDIC;
--
```

Figure 118. Sample output from generating SQL

```
-------------------------------------------------------------------------
-- Table space=DSNDB06.SYSALTER
-------------------------------------------------------------------------
--
  CREATE TABLESPACE SYSALTER
    IN DSNDB06
-- VCAT "00000001" -- DB2 catalog tablespace
    FREEPAGE 0 PCTFREE 7
    GBPCACHE CHANGED
    TRACKMOD YES
    LOGGED
    SEGSIZE 4
    BUFFERPOOL BP32K
    LOCKSIZE ROW
    LOCKMAX SYSTEM
    CLOSE NO
    COMPRESS NO
    CCSID       UNICODE
    DEFINE YES
    MAXROWS 255;
--
```

In some cases, data-partitioned secondary indexes might appear in the output because the process to generate the SQL supports these indexes.

Sample output with the Rebind option

If you specified that REBIND commands were to be generated on the Generate SQL from DB2 Catalog panel, not only is the preceding output displayed but so is the rebind output.

The following figure shows the rebind output.

```
  -------------------------------------------------------------------------------
  EDIT       SYS01311.T012717.RA000.ISTJE.R0215994         Columns 00001 00072
  Command ===>                                             Scroll ===> PAGE
  ****** **************************** Top of Data *****************************

  000001   REBIND PACKAGE(DSN8ES81.DSN8ES1)
  ****** *************************** Bottom of Data ***************************
```

Figure 119. Sample output of generating SQL with the REBIND option specified

Chapter 11. Running DB2 Admin performance queries

This information shows you how to run performance queries using DB2 Admin and describes the different types of performance queries that DB2 Admin supports.

Topics:
- "Option 1. Table Spaces Without RUNSTATS Information panel" on page 232
- "Option 1X. Indexes Without RUNSTATS Information panel" on page 234
- "Option 2. Table Spaces With More Than n Percent Relocated Rows panel" on page 235
- "Option 3. Indexes With Clustering Level Problems panel" on page 236
- "Option 4. Table Spaces With More Than n Percent Dropped Space panel" on page 238
- "Option 5. DB2 Table Spaces With Locking Size = 'S' panel" on page 239
- "Option 6. Indexes with 2 or More Levels panel" on page 240
- "Option 7. Indexes with 150 or more leaf page distance panel" on page 242
- "Option 8. Indexes On Tables With Fewer Than n Pages panel" on page 243
- "Option 9. Indexes Not Used By Any Plan or Package panel" on page 244
- "Option 10. Table Spaces Containing More Than One Table panel" on page 245
- "Option 11. Table Spaces Without SPACE Information panel" on page 246
- "Option 11X. Indexes Without SPACE Information panel" on page 247
- "Option 12. Table Spaces Exceeding Allocated Primary Quantity panel" on page 249
- "Option 12X. Indexes Exceeding Allocated Primary Quantity panel" on page 250
- "Option 13. Allocated and Used Space for Table Spaces panel" on page 251
- "Option 14. Table Space Maintenance Recommendations panel" on page 253
- "Option 14X. Index Space Maintenance Recommendations panel" on page 255
- "Option 15. Indexes not used within x number of days" on page 257

The DB2 Performance Queries panel (ADB23) is displayed when you select option 3 on the Administration Menu panel. Use this panel to select the DB2 performance and space utilization query you want to run. Select an option, and enter (part of) the name of the database for which the query should be run. See the descriptions that appear on each panel in this chapter for more information about each option shown in the following figure.

The select field on the performance queries panels lets you select an object, which is then shown on the corresponding system catalog panel. This lets you further investigate problems or choose to run utilities such as REORG and RUNSTATS.

```
ADB23 min ------------------ DB2 Performance Queries ------------------ 06:22
Option ===> _____


WHERE database LIKE  . . . _____              DB2 System: DSN9
AND obj has more than  . . 4     pages           DB2 SQL ID: ULVEMAN

    1  - Table spaces without RUNSTATS within 0    days   DB2 System: DB2X
    1X - Indexes without RUNSTATS within 0    days        DB2 SQL ID: ISTJE
 RUNSTATS information is required for options 2 through 9.
    2  - Table spaces with more than 10 percent relocated rows
    3  - Indexes with clustering level problems
    4  - Table spaces with more than 5 percent dropped space
    5  - Table spaces with locking size = 'S' (table space locking)
    6  - Index with 2  or more levels
    7  - Indexes with 150  or more leaf page distance
    8  - Indexes on tables with fewer than 6 pages
    9  - Indexes not used by any plan or package
   10  - Table spaces containing more than one table
   11  - Table spaces without SPACE information
   11X - Indexes without SPACE information
 SPACE information is required for options 12 through 13.
   12  - Table spaces exceeding allocated primary quantity
   12X - Indexes exceeding allocated primary quantity
   13  - Allocated and used space for table spaces
 RTS  Real-Time Statistics tables are required for options 14 and 14X.
   14  - Table Space maintenance recommendations
   14X - Index Space maintenance recommendations
   15  - Indexes not used within 40  days


                                    Switch Catalog Copy  . . . N (N/S/C)
```

Figure 120. DB2 Performance Queries panel (ADB23)

Option 1. Table Spaces Without RUNSTATS Information panel

The Table Spaces Without RUNSTATS Information panel is displayed when you select option 1 on the DB2 Performance Queries panel.

Tip: For table spaces that do not have RUNSTATS information, run the RUNSTATS utility on them.

The R line command enables you to quickly move to the Batch Job Utility Parameters panel (ADB2UPA). Entering the R line command is equivalent to entering S, UT, and R commands in succession.

The following figure shows the Table Spaces Without RUNSTATS Information panel.

```
DB2 Admin --- DB2X Table Spaces Without RUNSTATS    ROW 981 TO 1,000 OF 1,000
Command ===>                                                Scroll ===> PAGE

The following table spaces do not have RUNSTATS information. Consider running
the RUNSTATS utility on them.

Commands:      R - Runstats  UT - Utilities
Line commands: S - Select    R - Runstats

Select Name      Schema    DB Name   BP    L E S I C Ntable    N Active       Space
       *         *         *         *     * * * * *    *          *             *
------ --------  --------  --------  ----  - - - - - ------ ----------- -----------
       RGESI24S  RGET      RGED001   BP0   P N A N N    1          0             0
       RGESI26S  RGET      RGED001   BP0   P N A N N    1          0             0
       RGESMDAS  RGET      RGED001   BP0   P N A N N    1          0             0
       RGESM01S  RGET      RGED001   BP0   P N A N N    1          0             0
       RGESM02S  RGET      RGED001   BP0   P N A N N    1          0             0
       RGESOEGS  RGET      RGED001   BP0   P N A N N    1          0             0
       RGESOEIS  RGET      RGED001   BP0   P N A N N    1          0             0
       RGESOE0S  RGET      RGED001   BP0   P N A N N    1          0             0
       RGESOR1S  RGET      RGED001   BP0   P N A N N    1          0             0
       RGESOS1S  RGET      RGED001   BP0   P N A N N    1          0             0
```

Figure 121. Table Spaces Without RUNSTATS Information panel (ADB231)

The following fields are shown on this panel:

SELECT
Input field where you enter S to select a table space.

NAME
Name of the table space.

OWNER
Authorization ID of the owner of the table space.

DB NAME
Name of the database.

BP Name of the buffer pool used for the table space.

L Locking size, which is one of the following:
A Any
P Page
S Table space

E Erase rule, which is one of the following:
Y Erase
N No erase

S Status of the table space, which is one of the following:
A Available
N Not available

I Implicit (whether the table space was created implicitly), which is one of the
following:
Y Yes
N No

C Close rule, which is one of the following:
Y Yes
N No

NTABLE
Number of tables defined in the table space.

N ACTIVE

Number of active pages in the table space. This field is 0 if the RUNSTATS utility has not been run.

SPACE

Kilobytes (KB) of storage allocated to the table space. This field is 0 if the STOSPACE utility has not been run.

Option 1X. Indexes Without RUNSTATS Information panel

The Indexes Without RUNSTATS Information panel is displayed when you select option 1X on the DB2 Performance Queries panel.

Tip: For indexes that do not have RUNSTATS information, run the RUNSTATS utility on the indexes or on the table spaces using INDEX(ALL) option.

The R line command enables you to quickly move to the Batch Job Utility Parameters panel (ADB2UPA). Entering the R line command is equivalent to entering S, UT, and R commands in succession.

The following figure shows the Indexes Without RUNSTATS Information panel.

```
DB2 Admin -------- DB2X Indexes Without RUNSTATS Information ----- Row 1 of 54
Command ===>                                                 Scroll ===> PAGE

The following indexes do not have RUNSTATS information. Consider running the
RUNSTATS utility on the indexes or on the table spaces using INDEX(ALL).

Commands:       R - Runstats  UT - Utilities
Line commands:  S - Select     R - Runstats


                    Index                       Table
S Index Name        Schema    Table Name        Schema
  *                 *         *                 *
- ------------------ -------- ------------------ --------
  ADBCKPTX          ADB       ADBCHKPT          ADB
  JOBJECT_TABLE_IX  DBE       JOBJECT_TABLE     DBE
  OBJECT_TABLE_IX   DBE       OBJECT_TABLE      DBE
  DSN_REGISTER_APPLI DSNRGCOL DSN_REGISTER_APPL DSNRGCOL
  DSN_REGISTER_OBJTI DSNRGCOL DSN_REGISTER_OBJT DSNRGCOL
  XMAP_TBL          DSN8810   MAP_TBL           DSN8810
  XPARTS            DSN8810   PARTS             DSN8810
  CK0X              ISTFL2    CK0               ISTFL2
  TFLXLIM           ISTFL2    TFLTLIM           ISTFL2
  TFLXLIM2          ISTFL2    TFLTLIM2          ISTFL2
  TFLXLIM3          ISTFL2    TFLTLIM3          ISTFL2
  TFLXLIM4          ISTFL2    TFLTLIM4          ISTFL2
  TFLXLIM6          ISTFL2    TFLTLIM6          ISTFL2
  TFLXLTTX1         ISTFL2    TFLTLTTX1         ISTFL2
  TFLXLTTX2         ISTFL2    TFLTLTTX2         ISTFL2
  TFLXLTTX3         ISTFL2    TFLTLTTX3         ISTFL2
  TFLXLTTX4         ISTFL2    TFLTLTTX4         ISTFL2
  TFLXLTTX5         ISTFL2    TFLTLTTX5         ISTFL2
  TFLXNOVX1         ISTFL2    TFLTNOVX1         ISTFL2
  TFLXNOVY1         ISTFL2    TFLTNOVY1         ISTFL2
  TFLXV71           ISTFL2    TFLTV71           ISTFL2
  TFLXXXX           ISTFL2    TFLTXXX           ISTFL2
  TF2XLIM4          ISTFL2    TF2TLIM4          ISTFL2
  TF2XLIM5          ISTFL2    TF2TLIM5          ISTFL2
  XD                ISTFL2    TD                ISTFL2
  TYY_BX            ISTFL3    TYY               ISTFL3
  MAPX              ISTJE     MAP               ISTJE
  MAPX1             ISTJE     MAPT1             ISTJE
  MAPX2             ISTJE     MAPT2             ISTJE
```

Figure 122. Indexes Without RUNSTATS Information panel (ADB231X)

The following fields are shown on this panel:

S   Input field where you enter S to select an index.

INDEX NAME
        Name of the index.

INDEX SCHEMA
        Authorization ID of the schema of the index.

TABLE NAME
        Name of the table on which the index is defined.

TABLE SCHEMA
        Authorization ID of the schema of the table.

Option 2. Table Spaces With More Than n Percent Relocated Rows panel

The Table Spaces With More Than n Percent Relocated Rows panel is displayed
when you select option 2 on the DB2 Performance Queries panel.

You can change the percent argument by typing over it on the DB2 Performance Queries panel. The panel in the following figure shows 10 percent as an example.

Tip: For table spaces that have more than 10 percent relocated rows, that is, rows that are not located in their original page, reorganize the table spaces or review the pctfree and/or the free page values to leave more space for rows to grow during an update.

The O line command enables you to quickly move to the Batch Job Utility Parameters panel (ADB2UPA). Entering the O line command is equivalent to entering S, UT, and O commands in succession.

```
DB2 Admin ------- DB2X Table Spaces with Relocated Rows > 10 Pct --------------
Command ===>                                               Scroll ===> PAGE

The following table spaces have more than 10 percent relocated rows,
that is, rows not located in their original page. Consider reorganizing the
table spaces or redesigning the programs that update the rows.

Commands:        O - Reorg   UT - Utilities
Line commands:  S - Select   O - Reorg

  DB        TS                   Near        Far       Percent
S Name      Name       Part    Org Page   Org Page    Relocated        Rows
  *         *           *          *          *           *             *
- -------- -------- ------ ----------- ----------- ----------- -----------
  ISTJE2D  ISTJE2S     0         196          0          80           245
***************************** END OF DB2 DATA *********************************
```

Figure 123. Table Spaces With More Than n Percent Relocated Rows panel (ADB232)

The following fields are shown on this panel:

S   Input field where you enter S to select a table space.

DB NAME
    Name of the database.

TS NAME
    Name of the table space.

PART
    Partition number (zero if not partitioned).

NEAR ORG PAGE
    Number of rows that have been relocated near their original page.

FAR ORG PAGE
    Number of rows that have been relocated far from their original page.

PERCENT RELOCATED
    Percent of rows that have been relocated.

ROWS
    Number of rows in the table space or partition.

Option 3. Indexes With Clustering Level Problems panel

The Indexes With Clustering Level Problems panel is displayed when you select option 3 on the DB2 Performance Queries panel.

For indexes that have clustering level problems, the message `F.O.P TOO BIG` is displayed and indicates that the number of rows in a far offset position is greater

than 10 percent. In addition, `CLUSTERED xx` indicates that the index was defined as clustering, but the RUNSTATS utility found the clustering ratio to be less than 95 percent.

The O line command enables you to quickly move to the Batch Job Utility Parameters panel (ADB2UPA). Entering the O line command is equivalent to entering S, UT, and O commands in succession.

The following figure shows the Indexes With Clustering Level Problems panel.

```
DB2 Admin -------- DB2X Indexes with Clustering Level Problems ---- Row 1 of 1
Command ===>                                            Scroll ===> PAGE

The following indexes have clustering level problems. 'F.O.P TOO BIG' indicates
that the number of rows in a far offset position is greater than 10 percent.
'CLUSTERED xx' indicates that the index was defined as clustering but RUNSTATS
found the clustering ratio to be less than 95 percent. Consider reorganizing
the table spaces or redesigning your indexes, tables, and/or programs. Things
to consider are insert/update/delete patterns and frequencies, freespace/reorg
frequencies, and clustering sequences.


Commands:      O - Reorg   UT - Utilities
Line commands: S - Select   O - Reorg


                         Index     Pct in Far
S Index Name        Part  Schema   Offset Pos Clstrng Clstrd Comment
  *                  * *                 * *       *      *   *
- -----------------  ------ -------- ----------- ------- ------ ---------------
  XEMP2                  0 DSN8810          11 N       N      F.O.P TOO BIG
  DSNKAX01               1 V7COPY4          13 N       N      F.O.P TOO BIG
  DSNKAX03               1 V7COPY4          14 N       N      F.O.P TOO BIG
  DSNKDX02               0 V7COPY4          10 N       N      F.O.P TOO BIG
  ITEST                  1 V8DDHL1           0 Y       Y      CLUSTERED 80%
  ITEST2                 2 V8DDHL1           0 Y       Y      CLUSTERED 80%
****************************** END OF DB2 DATA ********************************
```

Figure 124. Indexes With Clustering Level Problems panel (ADB233)

The following fields are shown on this panel:

S   Input field where you enter S to select an index.

INDEX NAME
Name of the index.

PART
Number of partitions.

INDEX OWNER
Authorization ID of the owner of the index.

PCT IN FAR OFFSET POS
Percent of rows in a far offset position because of an insert into a full page.

CLUSTERING
Whether CLUSTER was specified when the index was created.

CLUSTERED
Whether the table is actually clustered by the index.

COMMENT
Reason why the index appears in the list.

Consider reorganizing the table spaces or redesigning your indexes, tables, and programs. Consider the insert/update/delete patterns and frequencies, freespace/reorganization frequencies, and clustering sequences.

Option 4. Table Spaces With More Than n Percent Dropped Space panel

The Table Spaces With More Than n Percent Dropped Space panel is displayed when you select option 4 on the DB2 Performance Queries panel.

You can change the percent argument by typing over it on the DB2 Performance Queries panel. The panel in the following figure shows 5 percent as an example.

When a table is dropped from a table space, the space it occupied cannot be reused. If the percent of dropped space is significant, consider reorganizing the table spaces and use segmented table spaces for the tables.

You should also run the MODIFY utility against table spaces that have dropped tables. Doing so removes the details of the table from the DBD.

The O line command enables you to quickly move to the Batch Job Utility Parameters panel (ADB2UPA). Entering the O line command is equivalent to entering S, UT, and O commands in succession.

The following figure shows the Table Spaces With More Than n Percent Dropped Space panel.

```
DB2 Admin ---- DB2X Table Spaces with More Than 5 Pct Dropped Space -----------
Command ===>                                              Scroll ===> PAGE

The following table spaces have more than 5 percent dropped space. When
a table is dropped from a table space, the space it occupied cannot be reused.
If the percentage of dropped space is significant, you should consider
reorganizing the table spaces and/or using segmented table spaces for the
tables.

Commands:       O - Reorg    UT - Utilities
Line commands:  S - Select    O - Reorg


                          Percent                   Primary Secondary
S DB Name  TS Name    Part Dropped        Rows    Quantity Quantity
  *        *          *  *                *           *         *
- -------- -------- ------ ------- ----------- ----------- ---------
  DSQ1STBB DSQ1STBT      0      10         135         100         5
  D208D001 D208SPRF      0      17         437           3         3
  D475D001 D475S088      0      94        8552          88        13
  D154D400 D154STPS      0      24         170           3         2
  D154D500 D154STEA      0      12           7         125         3
  D922D01  D922SINC      0      10          72           3         3
  JFDDB01  JFDSO4        0      39        1201         984       120
  JFDDB01  JFDSO5        0      20        2621        2280       240
```

Figure 125. Table Spaces With More Than n Percent Dropped Space panel (ADB234)

The following fields are shown on this panel:

S   Input field where you enter S to select a table space.

DB NAME
    Name of the database.

TS NAME
    Name of the table space.

PART
    Partition number (zero if not partitioned).

PERCENT DROPPED
Percent of space occupied by dropped tables.

CARD
Number of rows in the table space or partition.

PRIMARY QUANTITY
Primary space allocation in 4K blocks of storage.

SECONDARY QUANTITY
Secondary space allocation in 4K blocks of storage.

Option 5. DB2 Table Spaces With Locking Size = 'S' panel

The DB2 Table Spaces With Locking Size = 'S' panel is displayed when you select option 5 on the DB2 Performance Queries panel.

DB2 uses table space locking when accessing a table in the table space. Only use locking size = 'S' for read-only tables or tables that are accessed by only one user (or batch job) at a time. If concurrency between updating tasks or updaters and readers is required, then consider changing the locking size to 'A' (any locking) by altering the locksize with an ALTER SQL statement.

The AL line command enables you to quickly perform an ALTER TABLESPACE statement to change the LOCK SIZE to ANY. Entering the AL line command is equivalent to entering the S line command followed by the AL line command, and then entering ANY in the **LOCK SIZE** field.

The following figure shows the DB2 Table Spaces With Locking Size = 'S' panel.

```
DB2 Admin --------- DB2X Table Spaces with Locking Size = 'S'------------------
Command ===>                                                 Scroll ===> PAGE

The following table spaces have locking size = 'S'. DB2 will use table space
locking when accessing a table in the table space. You probably only want
locking size = 'S' for read-only tables or tables that are accessed by only one
user (or batch job) at a time. Consider changing the locking size to 'A' (any
locking), for example, by altering the locksize with an ALTER SQL statement.

 Commands:      UT - Utilities
 Line commands:   S - Select     AL - Alter

                     Lock Number of
 S  DB Name  TS Name Size    Tables
    *        *       *
 -- -------- -------- ---- ---------

    D402D10  D402SCIF  S        1
    D402D10  D402STIF  S        1
    D455D005 KBBSCOM   S        1
    D455D005 KBBSCTAB  S        1
    D455D005 KBBSIMS1  S        1
    D455D005 KBBSPRO   S        1
    D455D005 KBBSAPP   S        1
```

Figure 126. DB2 Table Spaces With Locking Size = 'S' panel (ADB235)

The following fields are shown on this panel:

S   Input field where you enter S to select a table space.

DB NAME
Name of the database.

TS NAME
　　Name of the table space.

LOCK SIZE
　　Lock size of the table space.

NUMBER OF TABLES
　　Number of tables defined in the table space.

Option 6. Indexes with 2 or More Levels panel

The Indexes with 2 or More Levels panel is displayed when you select option 6 on the DB2 Performance Queries panel.

You can specify the threshold for the number of levels (2 to 99).

The Indexes with 2 or More Levels panel shows the number of index levels. If the number exceeds 2 or 3, the performance of your application programs might suffer. Consider reorganizing the indexes more often or redesigning the indexes and tables. Consider key lengths, free space (pctfree and/or freepage), and insert/delete/update patterns and frequencies.

The O line command enables you to quickly move to the Batch Job Utility Parameters panel (ADB2UPA). Entering the O line command is equivalent to entering S, UT, and O commands in succession.

The following figures shows the Indexes with 2 or More Levels panel.

```
DB2 Admin -------- DB2X Indexes with 2 or More Levels  ----- Row 1 to 7 of 177
Command ===>                                               Scroll ===> PAGE

This panel shows indexes with 2  or more levels. If the number exceeds
2 or 3, it might have a negative impact on the performance of your
application programs. You might consider reorganizing the indexes more
often or redesigning the indexes and tables. Things to consider are
key lengths, free space, and insert/delete/update patterns and
frequencies.

Commands:      O - Reorg   UT - Utilities
Line commands:  S - Select   O - Reorg


                     Index                       Table   Index
S Index Name         Schema   Table Name         Owner   Levels
  *                  *        *                  *        *
- ---------------->  -------- ---------------->  -------- ------
  DSNDOB01           SYSIBM   SYSOBDS            SYSIBM    2
  DSNDOB02           SYSIBM   SYSOBDS            SYSIBM    2
  DSNUCX01           SYSIBM   SYSCOPY            SYSIBM    2
  IBMSNAP_PRUNCNTLXX ASN      IBMSNAP_PRUNCNTLXX ASN       2
  IBMSNAP_REGISTERXX ASN      IBMSNAP_REGISTERXX ASN       2
  XACT1              DSN8810  ACT                DSN8810   2
  XACT2              DSN8810  ACT                DSN8810   2
  XDEPT1             DSN8810  DEPT               DSN8810   2
  XDEPT2             DSN8810  DEPT               DSN8810   2
  XDEPT3             DSN8810  DEPT               DSN8810   2
  XEMP1              DSN8810  EMP                DSN8810   2
  XEMP2              DSN8810  EMP                DSN8810   2
  XEMPPROJACT1       DSN8810  EMPPROJACT         DSN8810   2
  XEMPPROJACT2       DSN8810  EMPPROJACT         DSN8810   2
  XPROJ1             DSN8810  PROJ               DSN8810   2
  XPROJ2             DSN8810  PROJ               DSN8810   2
  XPROJAC1           DSN8810  PROJACT            DSN8810   2
  XDSPTXT1           DSN8810  TDSPTXT            DSN8810   2
  XOPTVAL1           DSN8810  TOPTVAL            DSN8810   2
  TFLXLTT1           ISTFL2   TFLTLTT1           ISTFL2    2
  DSNFNX01           SYSIBM   LUNAMES            SYSIBM    2
  DSNOXX01           SYSIBM   SYSAUXRELS         SYSIBM    2
  DSNOXX02           SYSIBM   SYSAUXRELS         SYSIBM    2
  DSNSDX01           SYSIBM   SYSCHECKDEP        SYSIBM    2
  DSNSCX01           SYSIBM   SYSCHECKS          SYSIBM    2
  DSNCHX01           SYSIBM   SYSCHECKS2         SYSIBM    2
  DSNTNX01           SYSIBM   SYSCOLDIST         SYSIBM    2
  DSNHFX01           SYSIBM   SYSCOLDIST_HIST    SYSIBM    2
  DSNTPX01           SYSIBM   SYSCOLDISTSTATS    SYSIBM    2
```

Figure 127. Indexes with 2 or More Levels panel (ADB236)

The following fields are shown on this panel:

S Input field where you enter S to select an index.

INDEX NAME
Name of the index.

INDEX OWNER
Authorization ID of the owner of the index.

TABLE NAME
Name of the table on which the index is defined.

TABLE OWNER
Authorization ID of the owner of the table.

INDEX LEVELS
Number of levels in the index tree.

Option 7. Indexes with 150 or more leaf page distance panel

The Indexes with 150 or more Leaf Page Distance panel is displayed when you select option 7 on the DB2 Performance Queries panel.

You can specify the threshold for the leaf page distance (150 to 9999).

The leaf distance is defined as 100 times the average number of pages between successive leaf pages of the index. If this value exceeds 200, consider reorganizing the index. Also, consider redesigning the indexes. Consider free space/reorganization frequencies and insert/update/delete patterns and frequencies.

The O line command enables you to quickly move to the Batch Job Utility Parameters panel (ADB2UPA). Entering the O line command is equivalent to entering S, UT, and O commands in succession.

The following figure shows the Indexes with 150 or more Leaf Page Distance panel.

```
DB2 Admin---- DB2X Indexes with 150 or More Leaf Page Distanc Row 1 to 7 of 11
Command ===>                                               Scroll ===> PAGE

This panel shows indexes with 150 or more leaf page distance.  The
leaf page distance is defined as: 100 times the average number of pages
between successive active leaf pages of the index.If this value exceeds
200, consider reorganizing the index. You might also consider redesigning
the indexes. Things to consider are freespace/reorg frequencies and
insert/update/delete patterns and frequencies.

Commands:      O - Reorg   UT - Utilities
Line commands: S - Select   O - Reorg

                    Index                           Table        Leaf
S Index Name        Schema     Part Table Name      Schema       Distance
  *                 *          *  * *               *            *
- ----------------- -------- ------ ----------------- -------- -----------
  DSNAGH01          SYSIBM        0 SYSRESAUTH       SYSIBM          200
  DSNKAX01          SYSIBM        0 SYSPACKAUTH      SYSIBM          272
  DSNKAX02          SYSIBM        0 SYSPACKAUTH      SYSIBM          400
  DSNATX02          SYSIBM        0 SYSTABAUTH       SYSIBM          250
  DSNDCX01          SYSIBM        0 SYSCOLUMNS       SYSIBM          541
  DSNDKX01          SYSIBM        0 SYSKEYS          SYSIBM          184
  DSNHEX01          SYSIBM        0 SYSCOLUMNS_HIST  SYSIBM          385
  DSNKSX01          SYSIBM        0 SYSPACKSTMT      SYSIBM         1492
****************************** END OF DB2 DATA ******************************
```

Figure 128. Indexes with 150 or more Leaf Page Distance panel (ADB237)

The following fields are shown on this panel:

S   Input field where you enter S to select an index.

INDEX NAME
> Name of the index.

INDEX OWNER
> Authorization ID of the owner of the index.

PART
> Partition number (zero if not partitioned).

TABLE NAME
> Name of the table on which the index is defined.

TABLE OWNER
Authorization ID of the owner of the table.

LEAF DISTANCE
One hundred times the average number of leaf pages between successive active leaf pages of the index.

Option 8. Indexes On Tables With Fewer Than n Pages panel

The Indexes On Tables With Fewer Than n Pages panel is displayed when you select option 8 on the DB2 Performance Queries panel.

You can change the page number argument by typing over it on the DB2 Performance Queries panel. The panel in the following figure shows six pages as an example.

Consider dropping nonunique indexes that are defined on tables that have less than 6 pages. Unless the index is on a table in a table space that has multiple tables, it is unlikely to improve performance but will use resources to maintain its viability. However, do not drop unique indexes, indexes supporting constraints, clustering indexes, or the only index on a table without a full evaluation.

The DROP line command enables you to quickly issue a DROP INDEX statement. Entering the DROP line command is equivalent to entering an S line command and a DROP line command in succession.

The following figure shows the Indexes On Tables With Fewer Than n Pages panel.

```
 DB2 Admin ------- DB2X Indexes on Tables with Fewer Than 6 Pages  Row 30 of 38
 Command ===>                                              Scroll ===> PAGE

 The following nonunique indexes are defined on tables with less than 6
 pages. Such indexes defined on tables with less than 6 pages usually do not
 improve performance and should probably be dropped.

 Commands:      UT - Utilities
 Line commands:  S - Select      DROP - Drop Index


                      Index                     Table        Table
 Sel  Index Name      Schema   Table Name        Schema       Pages
      *               *        *                 *                *
 ---- ----------------- -------- ----------------> -------- -----------
      DSNTPX01        SYSIBM   SYSCOLDISTSTATS   SYSIBM           1
      DSNAUH01        SYSIBM   SYSUSERAUTH       SYSIBM           1
      DSNAUX02        SYSIBM   SYSUSERAUTH       SYSIBM           1
      XDEPT2          DSN8810  DEPT              DSN8810          1
      XDEPT3          DSN8810  DEPT              DSN8810          1
      XEMP2           DSN8810  EMP               DSN8810          2
      XPROJ2          DSN8810  PROJ              DSN8810          1
      XEMPPROJACT2    DSN8810  EMPPROJACT        DSN8810          1
      TFLXLTT1        ISTFL2   TFLTLTT1          ISTFL2           4
 ****************************** END OF DB2 DATA ******************************
```

Figure 129. Indexes On Tables With Fewer Than n Pages panel (ADB238)

The following fields are shown on this panel:

SEL
Input field where you enter S to select an index.

INDEX NAME
Name of the index.

INDEX OWNER
Authorization ID of the owner of the index.

TABLE NAME
Name of the table on which the index is defined.

TABLE OWNER
Authorization ID of the owner of the table.

TABLE PAGES
Total number of pages on which rows of the table appear.

Option 9. Indexes Not Used By Any Plan or Package panel

The Indexes Not Used By Any Plan or Package panel is displayed when you select option 9 on the DB2 Performance Queries panel.

Consider dropping indexes that are not used by any plan or package with static SQL if they are not used in QMF™ or any other dynamic SQL statement.

The DROP line command enables you to quickly issue a DROP INDEX statement. Entering the DROP line command is equivalent to entering an S line command and a DROP line command in succession.

The following figure shows the Indexes Not Used By Any Plan or Package panel.

```
DB2 Admin -------- DB2X Indexes Not Used by Any Plan or Package - Row 1 of 138
Command ===>                                               Scroll ===> PAGE

The following indexes are not used by any plan or package with static SQL.
Consider dropping the index if it is not used in QMF or any other dynamic SQL
statement.

Commands:       UT - Utilities
Line commands:   S - Select       DROP - Drop Index


                       Index                      Table
Sel  Index Name        Schema   Table Name        Schema
     *                 *        *                 *
---- ------------------ -------- ------------------ --------
     IBMSNAP_CRITSECX   ASN      IBMSNAP_CRITSEC    ASN
     IBMSNAP_PRUNCNTLX  ASN      IBMSNAP_PRUNCNTL   ASN
     IBMSNAP_REGISTERX  ASN      IBMSNAP_REGISTER   ASN
     IBMSNAP_SUBS_COLSX ASN      IBMSNAP_SUBS_COLS  ASN
     IBMSNAP_SUBS_EVENX ASN      IBMSNAP_SUBS_EVENT ASN
     IBMSNAP_SUBS_MEMBX ASN      IBMSNAP_SUBS_MEMBR ASN
     IBMSNAP_SUBS_SETX  ASN      IBMSNAP_SUBS_SET   ASN
     IBMSNAP_SUBS_STMTX ASN      IBMSNAP_SUBS_STMTS ASN
     IBMSNAP_UOW_IDX    ASN      IBMSNAP_UOW        ASN
     DSN_REGISTER_APPLI DSNRGCOL DSN_REGISTER_APPL  DSNRGCOL
     DSN_REGISTER_OBJTI DSNRGCOL DSN_REGISTER_OBJT  DSNRGCOL
     XACT1              DSN8810  ACT                DSN8810
     XACT2              DSN8810  ACT                DSN8810
     XDEPT1             DSN8810  DEPT               DSN8810
     XDEPT2             DSN8810  DEPT               DSN8810
     XDEPT3             DSN8810  DEPT               DSN8810
     XEMP1              DSN8810  EMP                DSN8810
     XEMP2              DSN8810  EMP                DSN8810
     XEMPPROJACT1       DSN8810  EMPPROJACT         DSN8810
     XEMPPROJACT2       DSN8810  EMPPROJACT         DSN8810
     XMAP_TBL           DSN8810  MAP_TBL            DSN8810
     XPARTS             DSN8810  PARTS              DSN8810
     XPROJ1             DSN8810  PROJ               DSN8810
     XPROJ2             DSN8810  PROJ               DSN8810
     XPROJAC1           DSN8810  PROJACT            DSN8810
     XCONA1             DSN8810  TCONA              DSN8810
     XDSPTXT1           DSN8810  TDSPTXT            DSN8810
     XOPTVAL1           DSN8810  TOPTVAL            DSN8810
```

Figure 130. Indexes Not Used By Any Plan or Package panel (ADB239)

The following fields are shown on this panel:

SEL
Input field where you enter S to select an index.

INDEX NAME
Name of the index.

INDEX OWNER
Authorization ID of the owner of the index.

TABLE NAME
Name of the table on which the index is defined.

TABLE OWNER
Authorization ID of the owner of the table.

Option 10. Table Spaces Containing More Than One Table panel

The Table Spaces Containing More Than One Table panel is displayed when you
select option 10 on the DB2 Performance Queries panel.

In general, nonsegmented table spaces should only contain one table. Unless you require more than one table per table space (for example, if you want to cluster small read-only tables in one table space), consider moving the tables to separate table spaces.

The following figure shows the Table Spaces Containing More Than One Table panel.

```
 DB2 Admin ----- DB2X Table Spaces Containing More Than One Table -- Row 1 of 6
 Command ===>                                              Scroll ===> PAGE

 The following nonsegmented table spaces contain more than one table. In most
 cases, nonsegmented table spaces should only contain one table. Unless you
 have good reasons for having more than one table per table space (for example,
 you want to cluster small read-only tables in one table space), consider moving
 the tables to separate table spaces.

 Commands:      UT - Utilities
 Line commands:   S - Select

                      Number of
 S DB Name  TS Name   Tables
   *         *              *
 - -------- --------- ---------
   DBEDB1    DBETS1         2
   DSN8D81A DSN8S81R       6
   DSQDBCTL DSQTSCT1        2
   DSQ1STBB DSQ1STBT       9
   ISTJED   ISTJES          6
   RAADB    RAATSQRC        2
 ****************************** END OF DB2 DATA ********************************
```

Figure 131. Table Spaces Containing More Than One Table panel (ADB2310)

The following fields are shown on this panel:

S   Input field where you enter S to select a table space.

DB NAME
    Name of the database.

TS NAME
    Name of the table space.

NUMBER OF TABLES
    Number of tables defined in the table space.

Option 11. Table Spaces Without SPACE Information panel

The Table Spaces Without SPACE Information panel is displayed when you select option 11 on the DB2 Performance Queries panel.

For table spaces that do not have SPACE information in the DB2 catalog, use the DB2 RUNSTATS and STOSPACE utilities to update the SPACE information. Consider running these utilities on a periodic basis. You can run RUNSTATS with options that just update the SPACE fields in the catalog.

The R line command enables you to quickly move to the Batch Job Utility Parameters panel (ADB2UPA). Entering the R line command is equivalent to entering S, UT, and R commands in succession.

The following figure shows the Table Spaces Without SPACE Information panel.

```
DB2 Admin ------ DB2X Table Spaces Without SPACE Information - Row 1 of 109
Command ===>                                                  Scroll ===> PAGE

The following table spaces do not have SPACE information in the DB2 Catalog.
The DB2 RUNSTATS and SPACE utilities can be used to update the SPACE
information. Consider running these utilities on a periodic basis.

Commands:        R - Runstats  SP - Stospace  UT - Utilities
Line commands:  S - Select     R - Runstats  SP - Stospace


                                 Storage  VSAM
S  DB Name  TS Name    Part Group      Catalog
   *         *          * *            *
-- --------  ---------  ------ --------  --------
   ADBDCH   ADBSCH          0 ADBGCH   DB2X
   DBEDB1   DBETS1          0 SYSDEFLT DB2X
   DBEDB2   DBETSSMP        0 SYSDEFLT DB2X
   DSNDB04  A               0 SYSDEFLT DB2X
   DSNDB04  AABC1OC9        0 SYSDEFLT DB2X
   DSNDB04  AABC1Z#Z        0 SYSDEFLT DB2X
   DSNDB04  CK0             0 SYSDEFLT DB2X
   DSNDB04  CK1             0 SYSDEFLT DB2X
   DSNDB04  DSNRFUNC        0 SYSDEFLT DB2X
   DSNDB04  DSNRSTAT        0 SYSDEFLT DB2X
   DSNDB04  MMRNAMES        0 SYSDEFLT DB2X
   DSNDB04  NAMES           0 SYSDEFLT DB2X
   DSNDB04  OBJECTRD        0 SYSDEFLT DB2X
   DSNDB04  PLANRTAB        0 SYSDEFLT DB2X
   DSNDB04  PLAN1$EE        0 SYSDEFLT DB2X
   DSNDB04  PLAN1GVH        0 SYSDEFLT DB2X
   DSNDB04  PLAN1PW#        0 SYSDEFLT DB2X
   DSNDB04  PLAN15TO        0 SYSDEFLT DB2X
   DSNDB04  SRP             0 SYSDEFLT DB2X
   DSNDB04  STAFF           0 SYSDEFLT DB2X
   DSNDB04  TD              0 SYSDEFLT DB2X
   DSNDB04  TESTSORT        0 SYSDEFLT DB2X
   DSNDB04  TESTSTUF        0 SYSDEFLT DB2X
   DSNDB04  TRI2            0 SYSDEFLT DB2X
   DSNDB04  TRI21PD3        0 SYSDEFLT DB2X
   DSNDB04  TYY             0 SYSDEFLT DB2X
   DSNDB04  T1              0 SYSDEFLT DB2X
   DSNDB04  T2              0 SYSDEFLT DB2X
   DSNDB04  UTLIST          0 SYSDEFLT DB2X
```

Figure 132. Table Spaces Without SPACE Information panel (ADB2311)

The following fields are shown on this panel:

S   Input field where you enter S to select a table space.

DB NAME
      Name of the database on which the table space resides.

TS NAME
      Name of the table space.

PART
      Partition number (zero if not partitioned).

STORAGE GROUP
      Name of the storage group for the table space.

VSAM CATALOG
      Name of the catalog used for space allocation.

Option 11X. Indexes Without SPACE Information panel

The Indexes Without SPACE Information panel is displayed when you select
option 11X on the DB2 Performance Queries panel.

For indexes that do not have SPACE information in the DB2 catalog, use the DB2 RUNSTATS and SPACE utilities to update the SPACE information. Consider running these utilities on a periodic basis.

The R line command enables you to quickly move to the Batch Job Utility Parameters panel (ADB2UPA). Entering the R line command is equivalent to entering S, UT, and R commands in succession.

The following figure shows the Indexes Without SPACE Information panel.

```
DB2 Admin -------- DB2X Indexes Without SPACE Information ----- Row 1 of 88
 Command ===>                                              Scroll ===> PAGE

 The following indexes do not have SPACE information in the DB2 Catalog.
 The DB2 RUNSTATS and SPACE utilities can be used to update the SPACE
 information. Consider running these utilities on a periodic basis.

 Commands:       R - Runstats  SP - Stospace  UT - Utilities
 Line commands:  S - Select    R - Runstats  SP - Stospace

                      Index          Storage VSAM
 S  Index Name        Schema   Part Group    Catalog
    *                 *        * *           *
 -- ----------------- -------- ------ -------- --------
    ADBCKPTX          ADB          0 ADBGCH   DB2X
    JOBJECT_TABLE_IX  DBE          0 SYSDEFLT DB2X
    OBJECT_TABLE_IX   DBE          0 SYSDEFLT DB2X
    DSN_REGISTER_APPLI DSNRGCOL    0 SYSDEFLT DB2X
    DSN_REGISTER_OBJTI DSNRGCOL    0 SYSDEFLT DB2X
    XMAP_TBL          DSN8810      0 DSN8G810 DB2X
    XPARTS            DSN8810      0 DSN8G810 DB2X
    CK0X              ISTFL2       0 SYSDEFLT DB2X
    TFLXLIM           ISTFL2       1 TFLSG    DB2X
    TFLXLIM           ISTFL2       2 TFLSG    DB2X
    TFLXLIM           ISTFL2       3 TFLSG    DB2X
    TFLXLIM2          ISTFL2       1 TFLSG    DB2X
    TFLXLIM2          ISTFL2       2 TFLSG    DB2X
    TFLXLIM2          ISTFL2       3 TFLSG    DB2X
    TFLXLIM3          ISTFL2       1 TFLSG    DB2X
    TFLXLIM3          ISTFL2       2 TFLSG    DB2X
    TFLXLIM3          ISTFL2       3 TFLSG    DB2X
    TFLXLIM3          ISTFL2       4 TFLSG    DB2X
    TFLXLIM3          ISTFL2       5 TFLSG    DB2X
    TFLXLIM4          ISTFL2       1 TFLSG    DB2X
    TFLXLIM4          ISTFL2       2 TFLSG    DB2X
    TFLXLIM4          ISTFL2       3 TFLSG    DB2X
    TFLXLIM4          ISTFL2       4 TFLSG    DB2X
    TFLXLIM4          ISTFL2       5 TFLSG    DB2X
    TFLXLIM6          ISTFL2       1 TFLSG    DB2X
    TFLXLIM6          ISTFL2       2 TFLSG    DB2X
    TFLXLIM6          ISTFL2       3 TFLSG    DB2X
    TFLXLIM6          ISTFL2       4 TFLSG    DB2X
    TFLXLIM6          ISTFL2       5 TFLSG    DB2X
```

Figure 133. Indexes Without SPACE Information panel (ADB2311X)

The following fields are shown on this panel:

S   Input field where you enter S to select an index.

INDEX NAME
Name of the index.

INDEX OWNER
Authorization ID of the owner of the index.

PART
Partition number (zero if not partitioned).

STORAGE GROUP
   Name of the storage group for the index.

VSAM CATALOG
   Name of the catalog used for space allocation.

Option 12. Table Spaces Exceeding Allocated Primary Quantity panel

The Table Spaces Exceeding Allocated Primary Quantity panel is displayed when you select option 12 on the DB2 Performance Queries panel.

For table spaces that exceed the allocated primary quantity, consider extending the primary allocation.

The AL line command enables you to quickly move to the Alter Table Space panel (ADB21SA). Entering the AL line command is equivalent to entering an S line command and then entering an AL line command.

The following figure shows the Table Spaces Exceeding Allocated Primary Quantity panel.

```
DB2 Admin ---- DB2X Table Spaces Exceeding Alloc Primary Quantity Row 14 of 30
Command ===>                                             Scroll ===> PAGE

The following table spaces exceed the allocated primary quantity. Consider
extending the primary allocation.

Note: If the primary or secondary quantity of 4K pages is less than the track
capacity for 4K blocks, then the number of extents shown is too high.

Commands:      UT - Utilities
Line commands:   S - Select      AL -Alter Tablespace


                            Primary Qty   Sec   Allocated   Pct Alloc
S  DB Name  TS Name   Part  (4K pages)    Qty   (4K pages) of Prim Qty    Ext
   *        *         *         *         *         *          *          *
-- -------- --------- ------ ----------- ------ ----------- ----------- ------
   DSNDB04  IBMS13#P    0          3       3         12         400        1
   DSNDB04  RAVN        0          3       3         36        1200        3
   DSNDB06  SYSSTR      0         72      72        144         200        2
   DSN8D81A DSN8S81D    0          8       5         12         150        1
   DSN8D81A DSN8S81E    1          3       3         36        1200        3
   DSN8D81A DSN8S81E    2          5       5         36         720        3
   DSN8D81A DSN8S81E    3          3       3         12         400        1
   DSN8D81A DSN8S81E    4          5       5         36         720        3
   DSN8D81A DSN8S81P    0         40      20         48         120        1
   DSN8D81P DSN8S81C    0         40      20         48         120        1
   ISTJED   ISTJES      0          3       3         12         400        1
   ISTJED   TDECP       1          3       3         12         400        1
   ISTJED   TDECP2      1          3       3         12         400        1
   TFLDB    TFLSLTT1    1          8       8         12         150        1
   TFLDB    TFLSLTT1    2          8       8         12         150        1
   TFLDB    TFLSLTT1    3          8       8         12         150        1
   TFLDB    TFLSLTT1    4          8       8         12         150        1
****************************** END OF DB2 DATA ******************************
```

Figure 134. Table Spaces Exceeding Allocated Primary Quantity panel (ADB2312)

The following fields are shown on this panel:

S   Input field where you enter S to select a table space.

DB NAME
   Name of the database.

TS NAME
Name of the table space.

PART
Partition number (zero if not partitioned).

PRIMARY QTY (4K PAGES)
Primary space allocation in 4K blocks of storage.

SEC QTY
Secondary space allocation in 4K blocks of storage.

ALLOCATED (4K PAGES)
Space allocated in 4K blocks of storage.

PCT ALLOC OF PRIM QTY
Percent of the primary quantity of space that is allocated.

EXT
Estimated number of extents for the table space.

Option 12X. Indexes Exceeding Allocated Primary Quantity panel

The Indexes Exceeding Allocated Primary Quantity panel is displayed when you select option 12X on the DB2 Performance Queries panel.

For indexes that exceed the allocated primary quantity, consider extending the primary allocation.

The AL line command enables you to quickly move to the Alter Index panel (ADB21XA). Entering the AL line command is equivalent to entering an S line command and then entering an AL line command.

The following figure shows the Indexes Exceeding Allocated Primary Quantity panel.

```
DB2 Admin --- DB2X Indexes Exceeding Alloc Primary Quantity  ROW 1 TO 9 OF 251
Command ===>                                        Scroll ===> PAGE

The following indexes exceed the allocated primary quantity. Consider extending
the primary allocation.

Note: If the primary or secondary quantity of 4K pages is less than the track
capacity for 4K blocks, then the number of extents shown is too high.

Commands:       UT - Utilities
Line commands:  S - Select       AL - Alter Index

   Index             Index          Prim Qty Sec Q  Allocated Pct Alloc
S  Name              Schema    Part (4K pgs)  (4K)  (4K pages) of Prim Q   Ext
   *                 *         *    *         *      *          *          *
-- ----------------> ------->  ---- -------- ------ ---------- --------- -----
   BKAXINC0          BKAT      1        250     25        288       115   3
   BKAXINC0          BKAT      2        225     23        240       106   2
   BKAXINC3          BKAT      0       1225    123       1320       107   2
   BKAXINC4          BKAT      0       3325    333       3420       102   2
   BKAXINC5          BKAT      0       1300    130       1452       111   3
   BKAXINC7          BKAT      0        250     25        252       100   2
   BKAXCUS0          BKAT      1        125     13        144       115   3
```

Figure 135. Indexes Exceeding Allocated Primary Quantity panel (ADB2312X)

The following fields are shown on this panel:

S   Input field where you enter S to select an index.

INDEX NAME
　Name of the index.

INDEX OWNER
　Authorization ID of the owner of the index.

PART
　Partition number (zero if not partitioned).

PRIM QTY (4K PGS)
　Primary space allocation in 4K blocks of storage.

SEC Q (4K)
　Secondary space allocation in 4K blocks of storage.

ALLOCATED (4K PAGES)
　Space allocated in 4K blocks of storage.

PCT ALLOC OF PRIM Q
　Percent of the primary quantity of space that is allocated.

EXT
　Estimated number of extents for the index.

Option 13. Allocated and Used Space for Table Spaces panel

The Allocated and Used Space for Table Spaces panel is displayed when you select option 13 on the DB2 Performance Queries panel.

The DB2 Performance Queries panel shows the allocated and used space for the table spaces in the databases you have selected. If the allocated space is much less than the used space, consider reducing the size of the table spaces.

The AL line command enables you to quickly move to the Alter Table Space panel (ADB21SA). Entering the AL line command is equivalent to entering an S line command and then entering an AL line command.

The following figure shows the Allocated and Used Space for Table Spaces panel.

```
DB2 Admin -------- DB2X Allocated and Used Space for Table Spaces Row 14 of 48
Command ===>                                               Scroll ===> PAGE

This panel shows the allocated and used space for the table spaces in the
databases you have selected. If the allocated space is much less than the used
space, consider reducing the size of the table spaces.

Note: If the primary or secondary quantity of 4K pages is less than the track
capacity for 4K blocks, then the number of extents shown is too high.

Commands:      UT - Utilities
Line commands:   S - Select      AL - Alter Tablespace

                        Prim Qty  Sec  Allocated      Pct      Pct
S  DB Name  TS Name   Part (in 4K)  Qty  (4K Pages)   Active  Dropped   Ext
   *         *                        *              *        *         *
-- -------- --------- ---- -------- ------ ---------- ---------- -------- ------
   DSNDB04  IBMS13#P    0        3      3         12          0        0      1
   DSNDB04  RAVN        0        3      3         36         34        0      3
   DSNDB06  SYSCOPY     0      540    540        540          0        0      1
   DSNDB06  SYSDBASE    0     3600   3600       3600         24        0      1
   DSNDB06  SYSDBAUT    0      132    132        132          4        0      1
   DSNDB06  SYSDDF      0      144    144        144          0        0      1
   DSNDB06  SYSGPAUT    0      144    144        144          2        0      1
   DSNDB06  SYSGROUP    0       48     48         48          0        0      1
   DSNDB06  SYSGRTNS    0      144    144        144          0        0      1
   DSNDB06  SYSHIST     0      144    144        144         38        0      1
   DSNDB06  SYSJAVA     0      144    144        144          0        0      1
   DSNDB06  SYSOBJ      0     1260   1260       1260          1        0      1
   DSNDB06  SYSPKAGE    0     1080   1080       1080         92        0      1
   DSNDB06  SYSPLAN     0     1260   1260       1260          8        0      1
   DSNDB06  SYSSEQ      0      144    144        144          0        0      1
   DSNDB06  SYSSEQ2     0      144    144        144          0        0      1
   DSNDB06  SYSSTATS    0     1620   1620       1620          1        0      1
   DSNDB06  SYSSTR      0       72     72        144         59        0      2
   DSNDB06  SYSUSER     0      108    108        108          4        0      1
   DSNDB06  SYSVIEWS    0     1800   1800       1800          6        0      1
   DSN8D81A DSN8S81D    0        8      5         12          1        0      1
   DSN8D81A DSN8S81E    1        3      3         36          1        0      3
   DSN8D81A DSN8S81E    2        5      5         36          0        0      3
   DSN8D81A DSN8S81E    3        3      3         12          1        0      1
   DSN8D81A DSN8S81E    4        5      5         36          0        0      3
```

Figure 136. Allocated and Used Space for Table Spaces panel (ADB2313)

The following fields are shown on this panel:

S   Input field where you enter S to select a table space.

DB NAME
    Name of the database.

TS NAME
    Name of the table space.

PART
    Partition number (zero if not partitioned).

PRIM QTY (IN 4K)
    Primary space allocation in 4K blocks of storage.

SEC QTY (4K PAGES)
    Secondary space allocation in 4K blocks of storage.

ALLOCATED (4K PAGES)
    Space allocated in 4K blocks of storage.

PCT ACTIVE
    Percent of the space that is occupied by rows of data from active tables.

PCT DROPPED
  Percent of the space this is occupied by rows of data from dropped tables.

EXT
  Estimated number of extents for the table space.

Option 14. Table Space Maintenance Recommendations panel

The Table Space Maintenance Recommendations panel is displayed when you select option 14 on the DB2 Performance Queries panel.

On this panel, you can enter values (or use the default values) that are used to calculate recommendations for actions to take. These recommendations can help you to determine when to run maintenance functions, such as COPY, REORG, or RUNSTATS on table spaces, or when to enlarge your DB2 data sets.

To use this option, real-time statistics tables are required to be present.

Restriction: The recommendations that DB2 Admin provides are based on general formulas and might not apply or be accurate for every installation. Further, if the real-time statistics tables contain only a small portion of information about your DB2 subsystem, the recommendations might not apply to the entire subsystem.

You can either enter parameters to be used in the formulas that query real-time statistics tables or you can use the defaults.

The following figure shows the Input Parameters for Real-Time Statistics panel.

```
DB2 Admin -------- DB2X Input Parameters for Real-Time Statistics ------- 09:39
Option ===>

The input values specified below are used in the calculations which determine
the recommended table space actions. For a full description of any parameter,
use panel HELP and refer to the entry indicated by the parenthesized keyword.

  Run using default settings:   (Yes/No)                       (default)
                                                                More:    +
  Limit, number of physical extents  . . . . . . . . . :   (50)
    (ExtentLimit)

  Limit, number of days since last image copy. . . . . :   (7)
    (CRDaySncLastCopy)

  Ratio, as percent, of updated pages to preformatted
    pages in table space or partition. . . . . . . . . :   (1)
    (CRUpdatedPagesPct)

  Ratio, as percent, of distinct updated pages to
    total active pages since last image copy . . . . . :   (1)
    (ICRUpdatedPagesPct)

  Ratio, as percent, of INSERTs, UPDATEs, DELETEs to
    total rows or LOBs since last full image copy. . . :   (10)
    (CRChangesPct)

  Ratio, as percent, of INSERTs, UPDATEs, DELETEs to
    total rows or LOBs since last incremental image
    copy . . . . . . . . . . . . . . . . . . . . . . . :   (1)
    (ICRChangesPct)

  Ratio, as percent, of INSERTs, UPDATEs, DELETEs
    to total rows or LOBs since last REORG . . . . . . :   (20)
    (RRTInsDelUpdPct)

  Ratio, as percent, of unclustered INSERTs to
    total rows or LOBs . . . . . . . . . . . . . . . . :   (10)
    (RRTUnclustInsPct)

  Ratio, as percent, of imperfectly chunked LOBs to
    total rows or LOBS . . . . . . . . . . . . . . . . :   (10)
    (RRTDisorgLOBPct)

  Ratio, as percent, of overflow records to total
    of rows or LOBs since last REORG or LOAD REPLACE . :   (10)
    (RRTIndRefLimit)

  Limit, number of mass deletes or dropped tables
    since last REORG or LOAD REPLACE . . . . . . . . . :   (0)
    (RRTMassDelLimit)

  Ratio, as percent, of INSERTs, UPDATEs, DELETEs
    to total rows or LOBs since last RUNSTATS. . . . . :   (20)
    (SRTInsDelUpdPct)

  Limit, sum of INSERTs, UPDATEs, DELETEs since
    last RUNSTATS. . . . . . . . . . . . . . . . . . . :   (0)
    (SRTInsDelUpdAbs)

  Limit, number of mass deletes since last REORG
    or LOAD REPLACE. . . . . . . . . . . . . . . . . . :   (0)
    (SRTMassDelLimit)

  Limit, number of times that data is accessed
    since last REORG or LOAD REPLACE . . . . . . . . . 0    (0)
    (REORGSCANACCESS)

  Limit, number of times that data is accessed using hash access
    since last REORG or LOAD REPLACE . . . . . . . . . 0    (0)
    (REORGHASHACCESS)




  Limit, number of bytes that were added or removed by UPDATE
    since the last REORG or LOAD REPLACE . . . . . . .       > (0)
    (UPDATESIZE)
```

Figure 137. Input Parameters for Real-Time Statistics panel (ADB2314T)

You can specify your own user values for the fields in the panel in the previous figure, and switch between these user values and the system default values. Use the RESET primary command to reset all user values to the system default values.

The Table Space Maintenance recommendations panel in the following figure shows a sample results panel that displays recommendations.

```
ADB2314 n ---------- DB2X Table Space Maintenance ------- Row 1 to 31 of 1,000
Command ===> _____  Scroll ===> PAGE
Max no of rows reached
Commands:       C - Full Copy  CI - Inc Copy  O - Reorg  R - Runstats
               (Add 'A' to primary commands to process all partitions
                in a single step, for example:  CA , CIA ,  OA ,  RA )
Line commands:  C - Full Copy  CI - Inc Copy  O - Reorg  R - Runstats
               AL - Resize   S - Select
                                       Pct   Num   <---Recommendations--->
Sel TSname   DBname    Part   Space(KB) Used  Ext   Copy Reorg Runst Resize
    *        *         *      _____ *    *     *    *     *     *
--- -------- -------- ------ ----------- ---- ------ ---- ----- ----- ------
___ DSN8S91E DSN8D91A  1400            ?    ?      ? FUL  YES   YES   NO
___ XPUR0000 DSN8D91X     0          720  100      1 FUL  YES   YES   NO
___ XSUP0000 DSN8D91X     0          720  100      1 FUL  YES   YES   NO
___ DSQTSRDO DSQDBCTL     0           48  100      1 FUL  YES   YES   NO
___ LI6510TS VNDS148      1           48  100      1 FUL  YES   YES   NO
___ LI6510TS VNDS148      2           48  100      1 FUL  YES   YES   NO
___ LI6510TS VNDS148      3           48  100      1 FUL  YES   YES   NO
___ LI6510TS VNDS148      4           48  100      1 FUL  YES   YES   NO
___ ARCHIVE1 DBADD101     0           48  100      1 FUL  YES   YES   NO
___ RETRIEV1 DBADD101     0           48  100      1 FUL  YES   YES   NO
```

Figure 138. Table Space Maintenance recommendations panel (ADB2314), which is the result of panel ADB2314T

Option 14X. Index Space Maintenance Recommendations panel

The Index Space Maintenance Recommendations panel is displayed when you select option 14X on the DB2 Performance Queries panel.

On this panel, you can enter values (or use the default values) that are used to calculate recommendations for actions to take. These recommendations can help you to determine when to run maintenance functions, such as COPY, REORG, or RUNSTATS on index spaces, or when to enlarge your DB2 data sets.

Requirement: To use this option, real-time statistics tables must be present.

Restriction: The recommendations that DB2 Admin provides are based on general formulas and might not apply or be accurate for every installation. Further, if the real-time statistics tables contain only a small portion of information about your DB2 subsystem, the recommendations might not apply to the entire subsystem.

You can either enter parameters to be used in the formulas that query real-time statistics tables or you can use the defaults.

The following figure shows the Input Parameters for Real-Time Statistics panel.

```
DB2 Admin -------- DB2X Input Parameters for Real-Time Statistics ------- 10:11
Option ===>

The input values specified below are used in the calculations which determine
the recommended index space actions. For a full description of any parameter,
use panel HELP and refer to the entry indicated by the parenthesized keyword.

  Run using default settings: YES (Yes/No)                    (default)
                                                                More:     +
  Limit, number of physical extents. . . . . . . . . : 50    (50)
    (ExtentLimit)

  Limit, number of days since last image copy. . . . . : 7    (7)
    (CRDaySncLastCopy)

  Ratio, as percent, of updated pages to preformatted
    pages. . . . . . . . . . . . . . . . . . . . . . : 1     (1)
    (CRUpdatedPagesPct)

  Ratio, as percent, of INSERTs, UPDATEs, DELETEs
    to total rows or LOBs since last image copy. . . . : 10   (10)
    (CRChangesPct)

  Limit, number of active pages. . . . . . . . . . . :      (50)
    (CRIndexSize)

  Ratio, as percent, of sum of inserted and deleted
    index entries to total since last REORG. . . . . :      (20)
    (RRIInsertDeletePct)

  Ratio, as percent, of inserted index entries with
    key greater than max to total since last REORG,
    REBUILD INDEX or LOAD REPLACE. . . . . . . . . . :      (10)
    (RRIAppendInsertPct)
  Ratio, as percent, of pseudo-deleted index entries
    to total since last REORG, REBUILD INDEX or
    LOAD REPLACE . . . . . . . . . . . . . . . . . . :      (10)
    (RRIPseudoDeletePct)

  Limit, number of mass deletes since last REORG,
    REBUILD, or LOAD REPLACE . . . . . . . . . . . . :      (0)
    (RRIMassDelLimit)

  Ratio, as percent, of number of index page splits
    far from original to total since last REORG,
    REBUILD INDEX or LOAD REPLACE. . . . . . . . . . :      (10)
    (RRILeafLimit)

  Limit, number of added or removed levels in index
    tree since last REORG, REBUILD INDEX, or LOAD
    REPLACE. . . . . . . . . . . . . . . . . . . . . :      (0)
    (RRINumLevelsLimit)

  Ratio, as percent, of number of inserted and deleted
    index entries to total since last RUNSTATS . . . . :      (20)
    (SRIInsDelUpdPct)

  Limit, number of inserted and deleted index entries
    since last RUNSTATS. . . . . . . . . . . . . . . :      (0)
    (SRIInsDelUpdAbs)

  Limit, number of mass deletes since last REORG,
    REBUILD INDEX or LOAD REPLACE. . . . . . . . . . :      (0)
    (SRIMassDelLimit)
```

Figure 139. Input Parameters for Real-Time Statistics panel (ADB2314I)

You can specify your own user values for the fields in the previous figure, and
switch between these user values and the system default values. Use the RESET
primary command to reset all user values to the system default values.

The Index Space Maintenance recommendations panel in the following figure shows a sample results panel that displays recommendations.

```
ADB23214X  ---------- DB2X Index Space Maintenance ---------- Row 1 to 13 of 13
Command ===>                                             Scroll ===> CSR

Commands:      C - Copy  O - Reorg  R - Runstats
Line commands: C - Copy  O - Reorg  R - Runstats  AL - Resize S - Select

    Index                                        <---Recommendations--->
Sel Space    DBname   Part Nactive    Space      Ext Copy Reorg Runst Resize
    *        *        *    *           *          *   *    *     *     *
--- -------- -------- ---- ----------- ----------- --- ---- ----- ----- ------
    AUXTST1X DSNDB04     0          12          48   1 YES  YES   YES   NO
    XCUSTLAS DSNDB04     0          12          48   1 YES  YES   YES   NO
    XCUST000 DSNDB04     0          12          48   1 YES  YES   YES   NO
    AUXBB31X DSNDB04     0          12          48   1 YES  YES   YES   NO
    SALE1FAM DSNDB04     0          12          48   1 YES  YES   YES   NO
    PLAN1L@B DSNDB04     0          12          48   1 YES  YES   YES   NO
    XTBIDENT DSNDB04     0          12          48   1 YES  YES   YES   NO
```

Figure 140. Index Space Maintenance recommendations panel (ADB2314X), which is the result of panel ADB2314I

Option 15. Indexes not used within x number of days

The Indexes panel is displayed when you select option 15 on the DB2 Performance Queries panel.

On this panel, the indexes that are not used within a range of days you select are shown. You can specify a range of days from 1 to 99999. The default for the number of days is 40. From this panel you can use ALT to alter the indexes.

```
ADB21X in -------------------- DSNA Indexes ------------ Row 1 to 29 of 1,000
Command ===>                                           Scroll ===> HALF
Max no of rows reached
Commands: DIS  STA  STO  ALL XSPACE
Line commands:
 T - Tables  D - Database  G - Storage group  P - Plans  C - Columns
 DIS - Display index space  STA - Start index space  STO - Stop index space
 ? - Show all line commands


                         Index                       Table        C C C C
Select Index Name        Schema    Table Name        Schema   U  Cols G D L M
       *                 *         *                 *        *     * * * * *
------ ----------------->--------  ------------------ -------- - ------ - - - -
       IB___C___DLQQI9X A       B___C___DLQQ8PC8 A        U     2 N N Y N
       IC___C___DLQ45RQ A       C___C___DLQ4PS6Y A        U     2 N N Y N
       IWK926A1          A540769 TWK926A1          A540769  U     2 N N N N
       IUADDC01          AD7CAQDC TBADDC01          AD7CAQDC P     1 N N N N
       IUADDC03          AD7CAQDC TBADDC03          AD7CAQDC P     1 N N N N
       IUADDC0C          AD7CAQDC TBADDC03          AD7CAQDC U     1 N N N N
       IUADDC0D          AD7CAQDC TBADDC0C          AD7CAQDC U     4 N N N N
       IUADDC2A          AD7CAQDC TBADDC2A          AD7CAQDC P     1 Y Y N N
       IUADDC2B          AD7CAQDC TBADDC2B          AD7CAQDC P     1 Y Y N N
       IXADDC01          AD7CAQDC TBADDC01          AD7CAQDC D     1 N N N N
       IXADDC03          AD7CAQDC TBADDC0C          AD7CAQDC D     1 N N N N
       IXADDC0A          AD7CAQDC TBADDC01          AD7CAQDC D     1 N N N N
       IXADDC2A          AD7CAQDC TBADDC2A          AD7CAQDC D     1 N N N N
       IXADDC2B          AD7CAQDC TBADDC2B          AD7CAQDC D     1 N N N N
       JWRDDC01_#_M4M    AD7CAQDC JWRDDC01          AD7CAQDC P     1 N N Y N
       ADBCHKX1          ADB      ADBCHK            ADB      U     4 N N N N
       ADBCKPTX          ADB      ADBCHKPT          ADB      P     3 N N Y N
       ADBHLDX1          ADB      ADBHOLD           ADB      U     4 N N N N
       ADBCHKX1          ADB10PAR ADBCHK            ADB10PAR U     4 N N N N
       ADBCKPTX          ADB10PAR ADBCHKPT          ADB10PAR P     3 N N Y N
       ADBHLDX1          ADB10PAR ADBHOLD           ADB10PAR U     4 N N N N
       ADB_GROUP_PROPERTY ADB3     ADB_PROPERTY      ADB3     U     3 N N N N
       ADB_PROPERTY_IDX  ADB3     ADB_PROPERTY      ADB3     D     2 Y Y N N
       ADB_PROPERTY_PK_ID ADB3    ADB_PROPERTY      ADB3     P     1 N N N N
       ADBCKPTX          ADB72PAR ADBCHKPT          ADB72PAR P     3 N N Y N
       IX_POLICY         ADEBOLT  POLICY_DATA       ADEBOLT  U     3 N N Y N
       IX_POLICY_STUFF   ADEBOLT  POLICY_STUFF      ADEBOLT  U     3 N N Y N
       I_DOCIDPURCHASEORD ADEBOLT PURCHASEORDERS    ADEBOLT  X     1 N N Y N
       I_NODEIDXPURCHASEO ADEBOLT XPURCHASEORDERS   ADEBOLT  N     4 Y Y Y N
```

Figure 141. Indexes not used within x number of days (ADB21X)

The following fields are shown on this panel:

S Input field where you enter S to select an index.

INDEX NAME
> Name of the index.

INDEX SCHEMA
> Authorization ID of the schema of the index.

TABLE NAME
> Name of the table on which the index is defined.

TABLE SCHEMA
> Authorization ID of the schema of the table.

Chapter 12. Using LISTDEFs and TEMPLATEs

LISTDEFs are used to specify multiple target objects either by specifying explicit names or patterns of names using wild cards, and TEMPLATEs allow you to define a data set pattern or mask to be used in place of JCL DD statements for various utilities.

A LISTDEF is a DB2 utility statement, which is used to group database objects into reusable lists. DB2 Utility processing generates a list of objects that matches the LISTDEF pattern or definition, and passes that list to the intended utility.

For example, if you want to make an image copy of all the table spaces in database ABC, you no longer need to explicitly list each table space. Instead, you can tell DB2 to make an image copy of every table space in database ABC. DB2 utility processing builds the list of objects during execution. The LISTDEF approach not only saves time, but also prevents an object from being erroneously omitted.

The TEMPLATE utility control statement eliminates the need for certain JCL DD statements during utility processing. In its simplest form, the TEMPLATE control statement defines the data set naming convention, but it can also control other allocation attributes, such as size or location.

TEMPLATE definitions can be used with or without LISTDEFs; therefore, within DB2 Admin the usage state for TEMPLATEs remains in effect for LISTDEF as well as non-LISTDEF utility invocations.

The template data set name is constructed during the processing of the utility, and is based on the template's data set name mask or pattern. The data set name mask or pattern is also likely to include an object-identifying pattern as part of its name, such as database or space name. Many variables can be used and combined together to form distinct and unique data set names. This allows a single utility job step to cover many data objects (using LISTDEFs, for example) and allows the target output data sets to be defined dynamically with TEMPLATEs.

DB2 Admin also supports the use of TEMPLATEs for DB2 Admin work data sets that are created and used in the jobs that are generated for the following functions: alter, restore, redefine, migrate, and object comparison. As with the utility data sets, TEMPLATES allow you to define your own data set naming convention and also control other allocation attributes for these non-utility work data sets. The set of variables that can be specified for the data set names for these non-utility work data sets depends on the DB2 Admin function.

For information on template types, see "Using user-defined or product default templates" on page 279

Note: Both DB2 Admin Tool and Object Comparison Tool support the use of REORG and COPY utilities in the Alter, OC, and CM functions. If COPYDDN 1 and COPYDDN 2 templates are specified, you should specify a unique symbolic variable to prevent conflicts.

Topics:
- "Managing LISTDEFs" on page 260
- "Managing TEMPLATEs" on page 270

Related tasks:

Instead of running utilities against explicitly specified table spaces or indexes, you might want to run the utilities against a predefined LISTDEF.

Managing LISTDEFs

With DB2 Admin, you can manage LISTDEFs by creating LISTDEF control tables, and by adding, editing, and deleting LISTDEFs.

Topics:

Creating the LISTDEF control tables

Before you can create and use LISTDEFs, you must create two DB2 control tables to store the LISTDEF definitions.

About this task

These tables have the following default names:
- DSNACC.UTLIST contains basic LISTDEF definitions.
- DSNACC.UTLISTE contains detailed LISTDEF definitions.

If you are using the DB2 Control Center, these tables might have already been created during installation by the DSNTIJCC.job. Before proceeding with the following steps, determine whether these tables already exist. If they do exist, go to "Editing a LISTDEF" on page 264.

To create the LISTDEF control tables:

Procedure

1. Select option 5 on the Administration Menu panel. The Utility generation using LISTDEFs and TEMPLATEs panel is displayed, as shown in the following figure.

```
ADB25 min ----- DSN9 Utility generation using LISTDEFs and TEMPLATEs ---- 00:33
Option ===>

  L - Manage LISTDEFs                                DB2 System: DSN9
  T - Manage TEMPLATEs                               DB2 SQL ID: ISTJE
 TU - Specify TEMPLATE usage

 CL - Create LISTDEF control table
 UL - Upgrade LISTDEF control table
 CT - Create TEMPLATE control table
 UT - Upgrade TEMPLATE control table

LISTDEF control table:
    Table owner . . . DSNACC   >
    Table name  . . . UTLIST            >

TEMPLATE control table:
    Table owner . . . DSNACC   >
    Table name  . . . UTTEMPLATE        >
```

Figure 142. Utility generation using LISTDEFs and TEMPLATEs panel (ADB25)

2. Select option CL. The LISTDEF Control Table panel is displayed, as shown in the following figure.

```
DB2 Admin ------------- DS2X LISTDEF Control Table---------------------- 11:05
Command ===>


Create LISTDEF control table

 Creator. . . . . DSNACC   > (optional, default is ISTJE)
 Name . . . . . . UTLIST           > (? to look up)

 IN
 Database . . . .           (optional, if blank DB2 implicitly creates a DB.?)
 Table space  . .           (optional, if blank DB2 implicitly creates a TS. ?))

 Index Creator  . . .          > (optional, default is ISTJE)
 Index Name . . . . . UTLSTX01          >
 Index Creator  . . .          > (optional, default is ISTJE)
 Index Name . . . . . UTLEX01           >
```

Figure 143. LISTDEF/TEMPLATE Control Table panel (ADB25C)

3. Specify the following values:
   - In the **Creator** and **Name** fields, specify a name for the control tables. Accept the default name (DSNACC.UTLIST) or enter a unique name. The control table that contains detailed LISTDEF definitions is automatically appended with an "E."

     Tip: Use the default name if you intend to use the DB2 Control Center in the future. Using the standard name eliminates the need to populate the DSNACC tables when you start using the Control Center. However, if you do choose the default name, be aware that running the DSNTIJCC job will drop any existing LISTDEF control tables.
   - In the **Database** and **Table Space** fields, specify location information for the control tables.
   - In the first set of **Index Creator** and **Index Name** fields, specify the name of the index creator and the name of the index for the basic LISTDEF definition table (DSNACC.UTLIST by default).
   - In the second set of **Index Creator** and **Index Name** fields, specify the name of the index creator and the name of the index for the detailed LISTDEF

Chapter 12. Using LISTDEFs and TEMPLATEs   **261**

definition table (DSNACC.UTLISTE by default). The index creator should match the name specified for the DSNACC.ULIST table, but the index creator name must be unique.

4. Press Enter to create the tables.

Upgrading the LISTDEF control tables

Use the UL command option to upgrade a LISTDEF control table to the current DB2 version.

About this task

To upgrade the LISTDEF control tables:

Procedure

1. Select option 5 on the Administration Menu panel. The Utility generation using LISTDEFs and TEMPLATEs panel is displayed.

```
ADB25 min ----- DSN9 Utility generation using LISTDEFs and TEMPLATEs ---- 00:33
Option ===>

  L - Manage LISTDEFs                            DB2 System: DSN9
  T - Manage TEMPLATEs                           DB2 SQL ID: ISTJE
 TU - Specify TEMPLATE usage

 CL - Create LISTDEF control table
 UL - Upgrade LISTDEF control table
 CT - Create TEMPLATE control table
 UT - Upgrade TEMPLATE control table

LISTDEF control table:
     Table owner . . . DSNACC   >
     Table name  . . . UTLIST            >

TEMPLATE control table:
     Table owner . . . DSNACE   >
     Table name  . . . UTTEMPLATE        >
```

Figure 144. Utility generation using LISTDEFs and TEMPLATEs panel (ADB25)

2. Select option UL on the option command line and press Enter. Validation of the table name is done to make sure it is a LISTDEF control table. The validation is based on the following column names and data types: NAME VARCHAR(18), TYPE VARCHAR(2), CREATEDBY VARCHAR(8), MODIFIEDBY VARCHAR(8), REMARKS VARCHAR(254). If the LISTDEF control table name is not at the current version, an upgrade is performed.

Adding a LISTDEF

Use the LISTDEFs panel to add a LISTDEF to the LISTDEF control tables.

About this task

To add a LISTDEF to the LISTDEF control tables:

Procedure

1. Select option 5 on the Administration Menu panel.
2. Select option L. The LISTDEFs panel is displayed, as shown in the following figure.

```
DB2 Admin --------- DB2X LISTDEFs in DSNACC.UTLIST -------- Row 1 to 17 of 17
Command ===>                                              Scroll ===> CSR

Line commands:
 A - Add  D - Delete  E - Edit UT - Utility selection
 U.x - Utility generation

Select Name              Creator  Type Remarks
       *                 *        *    *
------ ----------------- -------- ---- ------------------------------------->
       DBLT0301          SYSADM   B    linner, segmented and partitioned table
       DBLT0302          SYSADM   B    linner, segmented and partitioned table
       DBLT0303          SYSADM   B    linner, segmented and partitioned table
       DBLT0304          SYSADM   B    linner, segmented and partitioned table
       LISTLT03          SYSADM   B    dblt0301, dblt0302, dblt0303, and dblt0
       LT03I             SYSADM   I
       LT03T             SYSADM   T
       LT0301I           SYSADM   I
       LT0301T           SYSADM   T
       LT0302I           SYSADM   I
       LT0302T           SYSADM   T
       LT0303I           SYSADM   I
       LT0303T           SYSADM   T
       LT0304I           SYSADM   I
       LT0304T           SYSADM   T
       MYTABLES          DSNACC   B
       SYSIBM            DSNACC   T
****************************** END OF DB2 DATA *****************************
```

Figure 145. LISTDEFs panel (ADB25L)

The following fields are shown on this panel:

SEL   Input field where you enter the line command. The following line commands are valid:

   A   Add a new LISTDEF.

   D   Delete a LISTDEF.

   E   Edit a LISTDEF.

   UT   Invoke a utility against a LISTDEF.

   U.x Generate a utility job stream. Substitute 'x' with the LISTDEF utility option. For example, U.TU specifies use of a template for utility JCL and work statement list output.

NAME
         The name of a LISTDEF.

CREATOR
         Creator of the definition, or the last ID to update it.

TYPE   This field is included for compatibility with DB2 Control Center/390 (CC/390) and can have one of three values. For LISTDEFs that are added with DB2 Admin, the value is B (the default). This field is updatable.

   T   Table space

   I   Index space

   B   Both table space and index space

REMARKS
         This field contains an optional description of the LISTDEF. You can modify this field.

3. Issue the A line command. The Add LISTDEF panel is displayed, shown in the following figure.

```
DB2 Admin --------------- DB2X Utility LISTDEF - Add -------------------- 11:10
Command ===>

             Enter the following information:

  Name . . . .                    (LISTDEF name)
  Remarks  . .
```

Figure 146. Add LISTDEFs panel (ADB25LA)

The following fields are shown on this panel:

NAME
> Enter the name of the LISTDEF. This name must be unique for the control table being used.

REMARKS
> Enter an optional description of the LISTDEF.

4. Enter a unique name for the LISTDEF, identify the type of objects that the LISTDEF will apply to (T for table spaces, I for index spaces, or B for both) and optionally include a description of the LISTDEF.
5. Press Enter to add the LISTDEF to the LISTDEF control tables.

Editing a LISTDEF

With DB2 Admin, you can add, delete, or edit a clause contained in a LISTDEF.

About this task

Each LISTDEF consists of one or more clauses; each clause represents a separate line on the panel. When you initially define a LISTDEF, an empty clause is created. Use the following instructions to complete the definition of a new, empty clause, to edit an existing clause, or to delete a clause. You then fill in the fields to complete the definition of the clause; if you fail to fill in a required field, DB2 Admin prompts you for it. After a clause is created, you can edit it by typing over the field you wish to change or you can enter an E to the left of the clause to be changed. This latter approach can be used to edit a single clause.

Procedure

1. From the LISTDEFs panel, issue the E line command against the LISTDEF that you want to edit. The Edit LISTDEF panel is displayed, as shown in the following figure.

```
ADB25LE n -------------- DSNB Utility LISTDEF TEST1 ---------- Row 1 to 7 of 7
Command ===>                                                 Scroll ===> CSR

Line commands:
 A - Add  D - Delete  E - Edit  UT - Utility generation
 C - Copy
         Inc Targ Srch Obj   Srch Obj Srch Obj Name
Sel    # Exc Obj  Type       Qual     or Pattern      Cp Part Rel RI Cl Df H E
       * *   *    *          *        *               *  *    *   *  *  *  * *
--- ---- --- ---- ---------- -------- -------------- -- ----> --- - - - - -
       1 INC TBSP DATABASE            DB1             1     ALL          N
E      2 INC TBSP DATABASE            DB2             3                  Y
       3 INC TBSP DATABASE            DB003           15
       4 INC IXSP DATABASE            DB008           47:64
       5 EXC TBSP DATABASE            DB0107          14
       6 EXC TBSP TABLESPACE PEDRO    TS001           27
       7 EXC TBSP TABLE      PEDRO    TS003
****************************** END OF DB2 DATA ******************************
```

Figure 147. Edit LISTDEF control table panel (ADB25LE)

New, empty clauses are identified by a question mark (?) in the Incl/Excl field.

The following fields are shown on this panel:

SEL Action field where you enter the line command. The following line
commands are valid:

A Adds a new clause to the LISTDEF.

D Deletes a clause.

E Edits a LISTDEF clause. Use the Edit LISTDEF clause panel to
edit a single clause.

UT Invokes a utility against a single clause of the LISTDEF.

C Creates a copy of the selected clause.

The sequence number is part of a unique key which means that no two
clauses within the same LISTDEF can have the same sequence number.
The sequence of your clauses is important, because clauses are executed
in ascending order. If you need to reorder the clauses in a LISTDEF,
make room by updating the lowest clause that needs to be changed
with a sequence number greater than the others, then renumbering the
rest as needed.

INC/EXC
Include or exclude objects based on the search criteria. It is sufficient to
enter I or E.

TARG OBJ
This field refers to whether a list of table spaces or index spaces is to be
created. It is sufficient to enter T for table spaces or I for index spaces.

SRCH OBJ TYPE
This field refers to the type of object for which to search. The following
values are permissible:

D Database

L List

T Table

TS Table space

I or IX
>   Index

IS   Index space

SRCH OBJ QUAL
>   For object types table and index, this field indicates the owner.
>
>   For object types table space and index space, this field indicates the database name.
>
>   For certain object types, partial or complete wild-carding is available by using an asterisk (*). For example, DB01*.

SRCH OBJ NAME OR PATTERN
>   This field indicates the name of the search object, with partial or complete wild-carding available for certain object types. The wild card character is the asterisk (*).

CP   This field refers to COPY YES or COPY NO, and is applicable only to index spaces. For COPY YES, enter Y. For COPY NO, enter N.

Part   This field refers to the PARTLEVEL keyword, or, if a number is specified, to the partition that is to be included. Permissible values are:

>   **blank**
>   >   The PARTLEVEL keyword is not added to the LISTDEF clause. As a result, the entire set of partitions in a partitioned table space is included as one unit. A sample LISTDEF might look like this:

```
LISTDEF T -- 00000010 OBJECTS
  INCLUDE TABLESPACE R148286.DB2CLEAN
  INCLUDE TABLESPACE R148286.DSN8S81D
  INCLUDE TABLESPACE R148286.DSN8S81E
  INCLUDE TABLESPACE R148286.DSN8S81P
  INCLUDE TABLESPACE R148286.EMP1
  INCLUDE TABLESPACE R148286.PART
  INCLUDE TABLESPACE R148286.PLANRTAB
  INCLUDE TABLESPACE R148286.T1
  INCLUDE TABLESPACE R148286.T2
  INCLUDE TABLESPACE R148286.T3
```

>   **Y**   Each partition is included as a separate object; the result might look like this:

```
LISTDEF T -- 00000014 OBJECTS
  INCLUDE TABLESPACE R148286.DB2CLEAN
  INCLUDE TABLESPACE R148286.DSN8S81D
  INCLUDE TABLESPACE R148286.DSN8S81E PARTLEVEL(00001)
  INCLUDE TABLESPACE R148286.DSN8S81E PARTLEVEL(00002)
  INCLUDE TABLESPACE R148286.DSN8S81E PARTLEVEL(00003)
  INCLUDE TABLESPACE R148286.DSN8S81E PARTLEVEL(00004)
  INCLUDE TABLESPACE R148286.DSN8S81P
  INCLUDE TABLESPACE R148286.EMP1
  INCLUDE TABLESPACE R148286.PART PARTLEVEL(00001)
  INCLUDE TABLESPACE R148286.PART PARTLEVEL(00002)
  INCLUDE TABLESPACE R148286.PLANRTAB
  INCLUDE TABLESPACE R148286.T1
  INCLUDE TABLESPACE R148286.T2
  INCLUDE TABLESPACE R148286.T3
```

>   **1-4096**
>   >   Enter a single partition number in this range for it to be included. (For releases of DB2 prior to Version 8, the allowable range of values is 1 to 254.) The resultant LISTDEF might look like the following example:

```
              LISTDEF T -- 00000010 OBJECTS
                INCLUDE TABLESPACE R148286.DB2CLEAN
                INCLUDE TABLESPACE R148286.DSN8S81D
                INCLUDE TABLESPACE R148286.DSN8S81E PARTLEVEL(00001)
                INCLUDE TABLESPACE R148286.DSN8S81P
                INCLUDE TABLESPACE R148286.EMP1
                INCLUDE TABLESPACE R148286.PART PARTLEVEL(00001)
                INCLUDE TABLESPACE R148286.PLANRTAB
                INCLUDE TABLESPACE R148286.T1
                INCLUDE TABLESPACE R148286.T2
                INCLUDE TABLESPACE R148286.T3
```

integer1:integer2

Starting with DB2 Version 10, the partitions can be specified as a range. integer1:integer2 indicates the range of partitions to be specified in a list.

Rel   Auxiliary relationship can be ALL, BASE, LOB or XML. Specify one of the following values:

    A   Enter an A for ALL (base table spaces, related index spaces, and large objects).

    B   Enter a B for base table spaces and related index spaces.

    L   Enter an L for a large object.

    X   Enter an X for an XML object.

RI   Specify Y to include objects that are related through referential integrity.

CI   Filter the objects returned based on the existence or absence of cloned objects. The value can by Y or N

Df   Filter the LISTDEF objects based on whether data sets are defined or not. The value can be Y, N, A (all)

H   Specifies that only history objects should be included in the results.

E   Filter the objects returned by the LISTDEF based on the format of the RBA or LRSN.

    Y   Only objects with extended format are selected.

    N   Only objects with basic format are selected.

2. To edit existing clauses, you can either type over the field or fields that you want to change or you can issue the E line command to edit a single clause.

3. To add a clause, issue the A line command. A new empty clause, as identified by a question mark (?), is inserted, as shown in the panel in the following figure.

```
ADB25LE n --------- DB2X Utility LISTDEF A234567890123456 --- Row 1 to 1 of 1
Command ===>                                              Scroll ===> CSR

Line commands:
 A - Add  D - Delete  E - Edit  UT - Utility generation
 C - Copy
        Inc Targ Srch Obj  Srch Obj Srch Obj Name
Sel    # Exc Obj  Type     Qual     or Pattern      Cp Part  Rel RI Cl Df H E
       * *   *    *         *        *              *  *      *   *  *  *  * *
--- ---- --- ---- ---------- -------- -------------- -- ----> --- -  -  -  - -
       1 INC TBSP TABLESPACE DSNDB04  *
       2 INC IXSP TABLE      DSNDB04  *              Y
       3 ?
```

Figure 148. LISTDEF panel (ADB25LE) – adding a clause

4. Type in the fields to complete the definition of the clause and press Enter to complete the addition. Alternatively, you can enter an E to the left of the clause to bring up the Edit LISTDEF clause panel, which can be used to edit a single clause.

5. To delete a clause, issue the D line command against the clause that you want to delete.

Editing a single LISTDEF clause

Use the Edit LISTDEF clause panel to edit a single LISTDEF clause.

About this task

To edit a single LISTDEF clause:

Procedure

1. To display the Edit LISTDEF clause, issue the E line command against a LISTDEF. The following figure shows the Edit LISTDEF clause panel.

```
ADB25LEA  --------------- DSNB Utility LISTDEF - PSV1 ------------------- 17:58
 Command ===>


 Incl/Excl  . . . . . INCLUDE       (Include or Exclude)
 Target object  . . . TBSP          (TBSP or IXSP)
 Copy . . . . . . . .               (Yes/No)
 Srch object type . . DATABASE      (List, Database, TableSpace, IndexSpace,
                                      Table, Index)
 Srch object qual . . _____ > (Owner or Database to qualify NAME)
 Srch object name . . DB2_____ > (Name - Full or partial using *)
 PARTLEVEL  . . . . . 3                            > (Y, n, nnnn:mmmm)
 CLONED . . . . . . .               (Yes/No)
 DEFINED  . . . . . .               (Yes, No, ALL)
 RI related . . . . .               (Yes/No)
 Auxiliary
   relationship . . .               (All, Base, LOB or XML)
 HISTORY  . . . . . .               (Yes/No)
 Extended RBA . . . . YES           (Yes/No)


 Sequence . . . . . . 2             (Processing order)

 Press ENTER to update the LISTDEF clause.

 Statement  . . . : INCLUDE TABLESPACES  DATABASE DB2 PARTLEVEL(3)     EXTENDED
 YES
```

Figure 149. Edit LISTDEF clause panel (ADB25LEA)

2. Specify the following values. As you enter information in the fields, the generated LISTDEF clause is shown at the bottom of the panel.

   The following fields are shown on this panel:

   INCL/EXCL
   > Include or exclude objects based on the search criteria. It is sufficient to enter I include objects or E to exclude objects.

   TARGET OBJ
   > Permissible values are:
   >
   > **T**   Table space
   >
   > **I**   Index space

COPY This field refers to COPY YES or COPY NO, and is applicable only to index spaces. For COPY YES, enter `Y`. For COPY NO, enter `N`.

SRCH OBJ TYPE
This field refers to the type of DB2 Admin Look Up object for the initial search. The following values are permissible:

L List

D Database

TS Table space

IS Index space

TB Table

I or IX
Index

SRCH OBJ QUAL
For DB2 Admin Look Up types table and index, this field indicates the owner.

For DB2 Admin Look Up types table space and index space, this field indicates the database name.

For some DB2 Admin Look Up types, partial or complete wild-carding is available by using an asterisk (*). For example, DB01*.

SRCH OBJ NAME
This field indicates the name of the DB2 Admin Look Up object, with partial or complete wild-carding available for some DB2 Admin Look Up types. The wild card character is the asterisk (*).

PARTLEVEL
This field refers to the PARTLEVEL keyword, or, if a number is specified, to the partition that is to be included.

RI related
Specify `Y` to include objects that are related through referential integrity.

HISTORY
A filtering keyword that specifies that only history (versioning) objects should be included on the resulting list clause.

Extended RBA
Filter the objects returned by the LISTDEF based on the format of the RBA or LRSN.

- Yes - only objects with extended format are selected.
- No - only objects with basic format are selected.

Auxiliary relationship
This field indicates a large object type. Specify one of the following values:

A Specify an `A` for ALL (base table spaces, related index spaces, and large objects).

B Specify a `B` for base table spaces and related index spaces.

L Specify an `L` for LOB.

X Specify an `X` for XML.

Sequence

The sequence number is part of a unique key, which means that no two clauses with the same LISTDEF can have the same sequence number. The sequence of your clauses is important because they are executed in ascending order. If you need to reorder the clauses in a LISTDEF, create room by updating the lowest clause that needs to be changed with a sequence number greater than the others; then renumber the rest as needed.

Deleting a LISTDEF

Use the LISTDEFs panel to delete a LISTDEF from the LISTDEF control tables.

About this task

To delete a LISTDEF from the LISTDEF control tables:

Procedure

1. Select option 5 on the Administration Menu panel. The Utility generation using LISTDEFs and TEMPLATEs panel is displayed.
2. Select option L. The LISTDEFs panel is displayed.
3. Issue the D line command to delete the corresponding LISTDEF from the LISTDEF control tables.

Results

The LISTDEF is removed from the control tables.

Managing TEMPLATEs

With DB2 Admin, you can create and maintain TEMPLATEs.

Topics:
- "Adding, editing, or deleting a TEMPLATE"
- "Utility Template panel" on page 273
- "Utility Template — Dataset Name panel" on page 275

Adding, editing, or deleting a TEMPLATE

Use the TEMPLATEs panel to add, edit, or delete a TEMPLATE.

The TEMPLATEs panel, as shown in the following figure, is displayed when you select option T on the Utility generation using LISTDEFs and TEMPLATEs panel. The panel also displays when you enter a question mark (?) on the Utility Template Use panel. The TEMPLATEs panel presents the existing TEMPLATEs within the control table; the table name is shown in the panel header (in this case, DSNACC.UTTEMPLATE).

Use this panel to add, edit, or delete a TEMPLATE definition.

```
 DB2 Admin --------- DB2X TEMPLATEs in DSNACC.UTTEMPLATE --- Row 1 to 21 of 21
 Command ===>                                              Scroll ===> CSR

 Line commands: A - Add  E - Edit  D - Delete

 Sel   Name               Creator  Remarks
       *                  *        *
 ----- ------------------ -------- --------------------------------------------
 *     COPYLOC            SYSADM
       COPYREM            SYSADM
       COPYREM2           SYSADM
       FTERDDN            SYSADM
       INDDN              SYSADM
       SCOPY              SYSADM
       SCOPY2             SYSADM
       SCOPY3             SYSADM
       SCOPY4             SYSADM
       SCOPY5             SYSADM
       SDISC              SYSADM
       SERR               SYSADM
       SMAP               SYSADM
       SORTOUT            SYSADM
       SPUNCH             SYSADM
       SRCPY1             SYSADM
       SRCPY2             SYSADM
       SREC               SYSADM
       SUT1               SYSADM
       UNLDDN             SYSADM
       WORKDDN            SYSADM
 ***************************** END OF DB2 DATA *******************************
```

Figure 150. TEMPLATEs panel (ADB25T)

The following fields are shown on this panel:

SEL   Input field where you enter a line command. The following line commands
          are valid:

    A   Enter an A to add a new TEMPLATE.

    E   Enter an E to edit a TEMPLATE definition.

    D   Enter a D to delete a TEMPLATE.

    +   Enter a plus sign (+) to associate the template with a keyword on
          the Utility Template Use panel.

NAME
          This is the TEMPLATE name.

CREATOR
          Creator of the TEMPLATE, or the last ID to update it.

REMARKS
          This field contains an optional description of the TEMPLATE. You can
          modify this field.

Upgrading the TEMPLATE control tables

Use the UT command option to upgrade a TEMPLATE control table to the current
DB2 version.

About this task

To upgrade the TEMPLATE control tables:

Procedure

1. Select option 5 on the Administration Menu panel. The Utility generation using LISTDEFs and TEMPLATEs panel is displayed.

```
ADB25 min ----- DSN9 Utility generation using LISTDEFs and TEMPLATEs ---- 00:33
Option ===>

  L - Manage LISTDEFs                             DB2 System: DSN9
  T - Manage TEMPLATEs                            DB2 SQL ID: ISTJE
 TU - Specify TEMPLATE usage

 CL - Create LISTDEF control table
 UL - Upgrade LISTDEF control table
 CT - Create TEMPLATE control table
 UT - Upgrade TEMPLATE control table

LISTDEF control table:
    Table owner . . . DSNACC    >
    Table name  . . . UTLIST             >

TEMPLATE control table:
    Table owner . . . DSNACC    >
    Table name  . . . UTTEMPLATE       >
```

Figure 151. Utility generation using LISTDEFs and TEMPLATEs panel (ADB25)

2. Select option UT on the option command line and press Enter. Validation of the table name is done to make sure it is a TEMPLATE control table. The validation is based on the following column names and data types:
   - NAME VARCHAR(8),
   - CREATEDBY VARCHAR(8),
   - MODIFIEDBY VARCHAR(8),
   - DSN VARCHAR(254),
   - DISPSTATUS VARCHAR(3),
   - DISPNTERM VARCHAR(7),
   - DISPATERM VARCHAR(7),
   - DEVICETYPE VARCHAR(8),
   - MODELDCB VARCHAR(53),
   - BUFNO SMALLINT,
   - DATACLAS VARCHAR(8),
   - MGMTCLAS VARCHAR(8),
   - STORCLAS VARCHAR(8),
   - DSVOLSER VARCHAR(1784),
   - GDGLIMIT INTEGER,
   - EXPDL VARCHAR(10),
   - RETPD INTEGER,
   - UNITTYPE CHAR(1),
   - PQTY INTEGER,
   - SQTY INTEGER,
   - SPACEUNIT VARCHAR(3),
   - PCTPRIME INTEGER,
   - MAXPRIME INTEGER,
   - NBRSECND INTEGER,
   - UNCNT SMALLINT,

- STACK CHAR(1),
- JES3DD VARCHAR(8),
- TRTCH VARCHAR(6),
- REMARKS VARCHAR(254),
- VOLCNT SMALLINT.

If the TEMPLATE control table name is not at the current version, an upgrade is performed.

Utility Template panel

Use the Utility Template panel to add or edit a TEMPLATE utility control statement.

The Utility Template panel is displayed, as shown in the following figure, when you select option A or E on the TEMPLATEs panel. When adding a TEMPLATE, the input fields contain blanks. When editing a TEMPLATE, the previously stored values are displayed, which you can overwrite.

```
DB2 Admin -------------------- DB2X Utility Template -------------------- 11:20
Command ===>

Enter name and optional remark.  Press Enter to save.

TEMPLATE . . . .            (Template name)
Remark . . . . .
                                                                            >
Common options:
  UNIT . . . . .            (Device number, type or group name)
  Device type  .            (DASD or TAPE, default is DASD)
  DSN  . . . . .

  Change other common options  . .     (Yes/No)
  Change disk options  . . . . . .     (Yes/No)
  Change tape options  . . . . . .     (Yes/No)

Statement  . . . TEMPLATE
```

Figure 152. Utility Template panel (ADB25TE)

To create a new template, provide a TEMPLATE and a DSN and press Enter.

The following input fields are shown on this panel:

TEMPLATE
>Enter a name for the template. The template name must be unique within the control table that you are using.

REMARK
>Enter an optional description of the template.

UNIT Use this field to specify the device number or group name for the data set.

Device type
>Use this field to specify the device type for the data set.

DSN Use this field to provide a data set name pattern for the template. The data set name can be composed of variables whose value is determined and substituted during execution of the utility that is using the template or execution of the job that DB2 Admin generated for alter, restore, redefine, migrate, or object comparison processing that is using the template.

To construct a data set name pattern by using substitution variables, specify a question mark (?) as the first character of the **DSN** field. When you press Enter, the Utility Template — Dataset Name panel is displayed.

The variables displayed on the Utility Template — Dataset Name panel are the variables that are supported for normal DB2 utility template processing. Therefore, any variable displayed is valid for the data set name pattern for a utility data set template. However, not all of the variables are valid for the templates for non-utility work data sets, and additional variables might apply.

Change other common options

Use this field to specify additional attributes for the data set. When you specify `Yes` and press Enter, the Template Common Options panel (ADB25TC) is displayed, as shown in the following figure. See the online help for the description of the fields on this panel.

```
DB2 Admin ----------------- DB2X Template Common Options --------------- 11:21
Command ===>


  MODELDCB . . .
  BUFNO  . . . .              (Number of BSAM buffers)
  DATACLAS . . .              (SMS Data class)
  MGMTCLAS . . .              (SMS Management class)
  STORCLAS . . .              (SMS Storage class)
  RETPD  . . . .       or   EXPDL . . .
  VOLUMES( . . .                                             >  )
  VOLCNT . . . .              (Volume Count)
  GDGLIMIT . . .              (GDG Limit)
  DISP(  . . . .      ,          ,          )
```

Figure 153. Template Common Options panel (ADB25TC)

Change disk options

Use this field to specify additional options for the data set—those options that are applicable only to data sets that are on disk. When you specify `Yes` and press Enter, the Template Disk Options panel (ADB25TS) is displayed, as shown in the following figure. See the online help for the description of the fields on the panel.

```
DB2 Admin ------------------ DB2X Template Disk Options ----------------- 11:22
Command ===>


  SPACE( . . . .          ,         ) (Primary, Secondary)
               . .               (CYL TRK or MB)
  PCTPRIME . . .              (Percentage of space obtained as primary)
  MAXPRIME . . .              (Maximum allowable primary space allocation)
  NBRSECND . . .              (Number of secondary allocation divisions)
  DIR  . . . . .              (Directory blocks)
  DSNTYPE  . . .              (LIBRARY HFS PDS or NULL)
```

Figure 154. Template Disk Options panel (ADB25TS)

Change tape options

Use this field to specify additional options for the data set—those options that are applicable only for data sets on tape. When you specify `Yes` and press Enter, the Template Tape Options panel (ADB25TT) is displayed, as shown in the following figure. See the online help for the description of the fields on the panel.

```
DB2 Admin ----------------- DB8A Template Tape Options ---------------- 11:23
Command ===>


  UNCNT  . . . .              (Number of devices to allocate)
  STACK  . . . .              (Yes/No, Stack on same tape volumes)
  JES3DD . . . .              (JES3 DDname for tape allocation)
  TRTCH  . . . .              (Track recording technique - NONE COMP or NOCOMP)
```

Figure 155. Template Tape Options panel (ADB25TS)

Utility Template — Dataset Name panel

Use the Utility Template — Dataset Name panel to construct a template data set
name by selecting the substitution variables to use.

About this task

To construct a template data set name by selecting the substitution values to use:

Procedure

1. From the Utility Template panel, enter a ? in the **DSN** field. The Utility
   Template – Data Set Name panel is displayed, as shown in the following figure.

```
DB2 Admin ------------- DB2X Utility Template - Dataset Name ----------- 11:30
Command===>

Select symbolic variables or enter non-symbolic characters.  Processing for
this panel occurs in left to right, and top to bottom sequence.  Press ENTER
to process any current choices.

DSN Model . .

 Non-Symbolic characters  . .

 DB2 Symbolic Variables:

    JOBNAME  . . .    MVS  jobname     STEPNAME . . .    MVS step name
    UTILID . . . .    Utility ID       SSID . . . . .    Subsystem ID
    ICTYPE . . . .    Image Copy Type  UTILNAME . . .    Utility Name
    SEQ  . . . . .    Sequence Number  LOCREM . . . .    IC DDN usage
    PRIBAC . . . .    IC DDN Usage
    LIST . . . . .    List Name        DB . . . . . .    Database name
    TS . . . . . .    Table space      IS . . . . . .    Index Space
    SN . . . . . .    Space name       PART . . . . .    Part number (5-digit)
                                       DSNUM  . . . .    Part/piece number
    DATE . . . . .    YYYYDDD          TIME . . . . .    HHMMSS
    JDATE  . . . .    YYYYDDD          YEAR . . . . .    YYYY
    MONTH  . . . .    MM               DAY  . . . . .    DD
    JDAY . . . . .    DDD              HOUR . . . . .    HH portion of time
    MINUTE . . . .    MM portion       SECOND . . . .    SS portion of time
                                       UNIQ . . . . .    Unique identifier
    USERID . . . .    Batch userid

 DB2 Admin Symbolic Variables:

    PREFIX . . . .    Data set prefix  LEVEL  . . . .    Worklist name
    TNAME  . . . .    Table ID
```

Figure 156. Utility Template — Data Set Name panel (ADB25TD)

2. Specify substitution variables:
   - To specify non-symbolic characters, type them in the **Non-Symbolic
     characters** field. Press Enter to transfer and append the characters you
     entered to the DSN Model field near the top of the panel, which contains the
     template data set name pattern.

- To select a symbolic variable, type any character (such as a slash) to the right of the leader dots. Press Enter to transfer your choices to the DSN model, which causes the variable name, followed by either one or two periods, to be appended to the DSN model statement. The first period marks the end of the variable name, not the end of the qualifier. If the preceding item is a variable, two periods are required in succession to begin a new name segment (qualifier). The first period marks the end of the variable and the second period marks the beginning of the next qualifier.

  The variable names are appended to the data set name template in left-to-right and top-to-bottom order each time Enter is pressed. To append an earlier variable after a later variable, first select the later variable and press Enter; then append the earlier variable.

3. Verify that the data set name in the **DSN Model** field contains the appropriate number of periods. Also, for variables that will return numeric characters, ensure that an alphabetic character (A to Z) or national character (# @ $) precedes the variable if it begins a qualifier. Type directly in the field to make any changes.

   Restriction: Not all the symbolic variables that are listed are valid variables for the data set name pattern for the templates for DB2 Admin work data sets for alter, restore, redefine, migrate and object comparison processing, and additional variables might apply. To specify any additional variables that are not listed, use the **Non-Symbolic characters** field or type them directly into the **DSN Model** field.

Example

Example: `&JOBNAME..&STEPNAME.` displays two variables in succession. If the preceding item is a non-symbolic character and not a variable, only one period is used.

Example: In the example, `&JOBNAME.DSNCOPY`, no period follows DSNCOPY because it is the last qualifier and it is not a variable.

Example: In the example, `&USERID..D&DAY..M&MONTH..&DB(3,4).`, an alphabetic character precedes the variables DAY and MONTH because they return numeric characters. The use of substring notation on variables enables limiting the number of characters that are returned. Here, only four characters of the database name, starting at the third character, are returned.

Recommendation: Although it is permissible to enter variables in the DSN model by simply typing in the variables, use the panel fields to avoid spelling errors.

The example in the following figure uses the previous panel to show a partially completed DSN model statement; the non-symbolic TEST is about to be appended, followed by the *jobname* substitution variable.

```
ADB25TD n ------------ DB2X Utility Template - Data Set Name ----------- 11:32
Command===>

Select symbolic variables or enter non-symbolic characters.  Processing for
this panel occurs in left to right, and top to bottom sequence.  Press ENTER
to process any current choices.

DSN Model . .  &DB..&TS..&UTILID..D&DATE..H&HOUR.&MINUTE.

 Non-Symbolic characters  . . TEST

 DB2 Symbolic Variables:

   JOBNAME  . . . S MVS  jobname     STEPNAME . . .   MVS step name
   UTILID . . . .   Utility ID       SSID . . . . .   Subsystem ID
 .
 .
 .
```

Figure 157. Utility Template — Data Set Name example (ADB25TD) partial panel

Substitution variables in utility templates for PUNCHDDN

Typically, the template data set names for a utility are constructed by DB2 when the utility is processed, based on the template's data set name mask or pattern and substitution variables. However, when you use the DB2 Admin functions for alter (ALT), migrate, rename database, and object comparison, the data set name that is associated with PUNCHDDN for a utility is resolved fully at JCL build time.

The data set name must be fully resolved and have valid qualifiers when the JCL is built because the data set for PUNCHDDN also becomes the input to the LOAD utility as the //SYSIN DD card. However, when the JCL is built for the data set name for PUNCHDDN, the value of some variables is unknown, and placeholder values are used instead. For example, if &JO or &JOBNAME is used as a substitution variable, JOBNAME is used as the value in the data set name.

The following table shows the replacement values for the symbolic variables that cannot be resolved at JCL build time for PUNCHDDN for (ALT), migrate, rename database, and object comparison:

Table 10. Replacement values for symbolic variables for templates for PUNCHDDN. Replacement values for symbolic variables for templates for PUNCHDDN

Symbolic variable	Replacement value
JOBNAME or JO	JOBNAME
UTILID	UTILID
STEPNAME	STEPNAME
SSID	The SSID
ICTYPE	ICTYPE
SEQ	SEQ
PRIBAC	PRIBAC
UTILNAME	UTILNAME
LOCREM	LOCREM
LIST	LIST
TS	The table space

Table 10. Replacement values for symbolic variables for templates for PUNCHDDN (continued). Replacement values for symbolic variables for templates for PUNCHDDN

Symbolic variable	Replacement value
SN	The table space
DB	The database name
IS	IS
PART	ALL
DATE	Build date in form YYYYDDD, for example, 2014190
JDATE	Julian date. Build date in form YYYYDDD, for example, 2014190
MONTH	The month, for example, 07
JDAY	The Julian day, for example, 190
MINUTE	The minutes, for example, 54
TIME	The time HHMMSS, for example, 135433
YEAR	The year, for example, 2014
DAY	The day, for example, 09
HOUR	The hour, for example, 13
SECOND	The seconds, for example, 33
USERID	The userid

TEMPLATE usage

You can associate a template with a particular data set—either a DB2 utility data set or a DB2 Admin work data set.

About this task

Many DB2 utilities use templates for certain ddnames used by the utility. The DB2 utilities that support the use of templates do so via a ddname keyword clause. For example, REORG TABLESPACE has a WORKDDN() keyword. The WORKDDN entries in the Template Usage panel correspond to any utility with the WORKDDN clause that supports templates. Certain keywords allow two parameters, such as WORKDDN for REORG TABLESPACE. The 'keyword 1' entry corresponds to the first subparameter for the keyword, while 'keyword 2' corresponds to the second subparameter.

The DB2 Admin work data sets that support the use of templates do so via a template keyword. For example, the work data set that the DB2 Admin Alter ALT function uses for the DDL that is extracted from the catalog is ALDDL.

To associate a template with the ddname keyword of a utility data set or template keyword of a non-utility work data set:

Procedure

1. Issue the TU (Template Usage) option with utility generation on the LISTDEFs and TEMPLATEs (ADB25) panel. The Specify UTILITY TEMPLATE Usage panel that is similar to the panel that is shown in the following figure is

displayed. The panel contains a list of keywords and columns showing whether a template is actively associated with that keyword, the name of the template, and the template's comment.

Note: Panel ADB25TU4 is used for the CLONE template type.

```
ADB25TU3 ------------------ DB2X Specify UTILITY TEMPLATE Usage --------- 11:45
Command ===>

Line commands:
 T - Toggle Use On/Off   C - Clear data   ? - Choose Template for the Keyword
 E - Edit Template
 Template type . . . . . . UTIL      (UTIL,CHG,MIG,MISC,CLONE)
 Generate templates  . . . YES       (Yes/No)
Sel Keyword      Use Template Comment
--- ----------- --- -------- -----------------------------------------------
    COPYDDN    1 /  SCOPY
    COPYDDN    2 /  COPYLOC
    DISCARDDN    /  COPYREM
    ERRDDN       /  COPYREM
    FILTERDDN    /  COPYREM2
    INDDN        /  COPYREM2
    MAPDDN       /  COPYREM
    PUNCHDDN     /  SPUNCH
    RECOVERYDDN1 /  COPYLOC
    RECOVERYDDN2 /  SRCPY1
    UNLDDN       /  UNLDDN
    WORKDDN    1 /  WORKDDN
    WORKDDN    2 /  SORTOUT
    LOBCOLDDN    /  CLOBDD
    XMLCOLDDN    /  CXMLDD
```

Figure 158. Specify UTILITY TEMPLATE Usage panel (ADB25TU3)

2. To change the list of template keywords and keyword associations that are displayed, overwrite the value in the **Template Type** field and press Enter. The following values are permissible:

UTIL Utility data set keywords used by DB2 utilities

CHG Alter non-utility data set keywords used by DB2 Admin Alter (ALT) function, DB2 Object Comparison Tool, or Change Management

MIG Migrate data set keywords used by the DB2 Admin Migrate function

MISC SYSPRINT data set keywords used by DB2 Admin for generating work statement lists (WSLs) online

CLONE
Utility templates used for cloned table spaces.

3. Enter ? in the **SEL** field and press Enter to associate a template with a keyword. The Templates panel that shows a list of defined templates is displayed.

4. Select a template by entering a plus sign (+) next to its name and pressing Enter. Figure 158 is displayed again with a slash (/) in the **Use** field and with the template name and its associated comment in the other two columns. A slash in the USE column for a keyword indicates a TEMPLATE statement will be built for any utility supporting templates for that keyword.

Using user-defined or product default templates

There are two types of templates you can use: templates that you specify yourself, and product default templates.

User-defined template

Template that you specify. User-defined templates can be found in the ADBTEMPL DD data definition. Refer to "Symbol variables in the ADBTEMPL file: DB2 TEMPLATE support" on page 713 for information about using symbol variables to specify DB2 TEMPLATE statements.

Product default template

Template assigned by DB2 Admin if you do not specify a template.

If you use a product default template, you need to manually add the --#TEMPLATE comment statement in the WSL. For example, if the MAPDDN template is defined, add the following comment statement:

```
--#TEMPLATE UTLMAP TYPE(TAPE)
TEMPLATE UTLMAP DSN 'SYSADM.XXX.T001'
UNIT TAPE
```

If the user-defined templates WORKDDN, MAPDDN, and ERRDDN are on removal media devices, you do not need to add the SPACE keyword.

Using the utility template to unload data from LOBs

If you want to unload data from a LOB column, you should use a utility template.

When a table that contains multiple LOB columns needs to be unloaded, each LOB column requires a partitioned data set (PDS). You can use any utility-supported variables to define this template. The variables must be unique to ensure that data is not overwritten during unloads. If you do not specify a template, the functions (such as ALT and MIG) will use the default template that DB2 Admin assigns.

The utility template for LOBs is used as follows:

1. The function (such as ALT and MIG) generates the unload.
2. The utility template statements are added to the WSL.
3. The WSL runs, and ADBTEP2 converts the UNLOAD syntax before passing it to DB2.

To set up and use the utility template for LOBs, follow the steps in "TEMPLATE usage" on page 278. After you have associated the template name with the LOBCOLDDN keyword, the following panel is displayed.

```
 DB2 Admin ------------ DB2X Specify UTILITY TEMPLATE Usage ----- 11:45
  Command ===>

  Line commands:
   T - Toggle Use On/Off   C - Clear data   ? - Choose Template for the Keyword
   E - Edit Template
   Template type . . . . . . UTIL      (UTIL,CHG,MIG,MISC,CLONE)
   Generate templates  . . . YES       (Yes/No)
 Sel Keyword      Use Template Comment
 --- ------------ --- -------- ------------------------------------------------
     COPYDDN    1  /  SCOPY
     COPYDDN    2  /  COPYLOC
     DISCARDDN     /  COPYREM
     ERRDDN        /  COPYREM
     FILTERDDN     /  COPYREM2
     INDDN         /  COPYREM2
     MAPDDN        /  COPYREM
     PUNCHDDN      /  SPUNCH
     RECOVERYDDN1  /  COPYLOC
     RECOVERYDDN2  /  SRCPY1
     UNLDDN        /  UNLDDN
     WORKDDN    1  /  WORKDDN
     WORKDDN    2  /  SORTOUT
     LOBCOLDDN     /  LOBTMPL1
     XMLCOLDDN
```

Figure 159. Specify UTILITY TEMPLATE Usage panel (ADB25TU)

Notes:

- The ADBL prefix is reserved for LOB template names that will be generated by the Run WSL function.
- The LOBCOLDDN data set name cannot exceed 35 bytes and must be a PDS.
- Do not specify a member name (for example, ADB.TEST.LOBCOL.OUT(MEMB2)).

Using the utility template to unload data from an XML column

If you want to unload data from an XML column, you should use a utility template.

When a table that contains multiple XML columns needs to be unloaded, each XML column requires a partitioned data set (PDS). You can use any utility-supported variables to define this template. The resulting data set name that is built using the variables must be unique to ensure that data is not overwritten. If you do not specify a template, the functions (such as ALT and MIG) will use the default template that DB2 Administration Tool assigns.

The utility template for XML is used as follows:

1. The function (such as ALT and MIG) generates the unload.
2. The utility template statements are added to the WSL.
3. The WSL runs, and ADBTEP2 converts the UNLOAD syntax before passing it to DB2.

To set up and use the utility template for XML data, follow the steps in "TEMPLATE usage" on page 278. After you have associated the template name with the XMLCOLDDN keyword, the following panel is displayed.

```
DB2 Admin -----------DB2X Specify UTILITY TEMPLATE Usage ----------- 11:45
 Command ===>

 Line commands:
  T - Toggle Use On/Off   C - Clear data    ? - Choose Template for the Keyword
  E - Edit Template
  Template type . . . . . . UTIL      (UTIL,CHG,MIG,MISC,CLONE)
  Generate templates  . . . YES        (Yes/No)
 Sel Keyword      Use Template Comment
 --- ------------ --- -------- ----------------------------------------------
     COPYDDN    1 /  SCOPY
     COPYDDN    2 /  COPYLOC
     DISCARDDN    /  COPYREM
     ERRDDN       /  COPYREM
     FILTERDDN    /  COPYREM2
     INDDN        /  COPYREM2
     MAPDDN       /  COPYREM
     PUNCHDDN     /  SPUNCH
     RECOVERYDDN1 /  COPYLOC
     RECOVERYDDN2 /  SRCPY1
     UNLDDN       /  UNLDDN
     WORKDDN    1 /  WORKDDN
     WORKDDN    2 /  SORTOUT
     LOBCOLDDN    /  LOBTMPL1
     XMLCOLDDN    /  XMLTMPL1
```

Figure 160. Specify UTILITY TEMPLATE Usage panel (ADB25TU)

Notes:

- The ADBX prefix is reserved for XML template names that will be generated by the Run WSL function.
- The XMLCOLDDN data set name cannot exceed 35 bytes and must be a PDS.
- Do not specify a member name (for example, ADB.TEST.XMLCOL.OUT(MEMB2)).

Chapter 13. Changing DB2 objects

With DB2 Admin, you can change a database, table space, table, index, or view.

Topics:

Overview of changing objects in DB2 Admin

With DB2 Admin, you can change a database and other objects such as table spaces, tables, indexes, or views.

For certain changes that are supported by the DB2 ALTER statement, DB2 Admin uses a DB2 ALTER statement to make the changes.

You can use the line commands AL and ALT to change DB2 objects.
- You use AL with a specified object type. When you use AL line command, the results of the procedure are SQL ALTER statements.
- ALT allows more changes to be made and more objects to be included. Also, with ALT, you can run utilities.

Examples of AL or ALT are as follows:

Table 11. Some examples of using AL or ALT to change objects

Action	Panel
Change aliases (ALT only)	ADB21A
Change databases	ADB21D
Change functions	ADB21F
Change triggers	ADB21J
Change stored procedures	ADB21O
Change sequences (ALT does not include types I and X)	ADB21Q
Change sequence aliases	ADB21Q
Change table spaces	ADB21S
Change tables	ADB21T
Change views (ALT only)	ADB21T
Change indexes	ADB21X
Change synonyms (ALT only)	ADB21Y
Change global variables (ALT only)	ADB21GV
Change foreign keys (ALT only)	ADB21TFK

ALT triggers the appropriate change dialog for the object type. When the dialog completes, the Alter Tables panel (ADB27CA) appears. This panel is the hub of the ALTER process. Here you can add objects, for example, by using the REL line command against a table. You can also add objects using the ADD primary command.

Use the primary command ALTER on the Alter Table (ADB27CA) panel to invoke analysis processing.

You can choose to perform analysis in batch by choosing **Perform analysis in batch (YES)** on the ALTER Analysis Options panel (ADBP7P). With this choice, the ALTER - Build Analyze and Apply Job panel (ADBPALT) panel appears. On this panel you can choose options for building the WSL or batch job used to implement the change.

You can choose to perform online analysis by entering **Perform analysis in batch (NO)** on the ALTER Analysis Options panel (ADBP7P). If the analysis process determines that SQL ALTER statements accomplish the task, panel ADB27CTC is then presented for you to choose to perform the SQL statements in foreground (online) or to generate a batch job. If ALTER statements are chosen, the SQL is performed. If batch jobs are chosen, then panel ADBPALT is displayed. Also, if the analysis process determines that a DB2 ALTER statement cannot be used, then panel ADBPALT is displayed.

After DB2 Admin generates the batch jobs, you can review them and then submit them to perform the changes.

You can use the Batch Restart program, ADBTEP2, to restart or resume the execution of an Alter job at an intermediate point, if one of the SQL statements in the input stream fails. In addition, you can combine the generated Alter batch jobs into a single job.

Changing databases

You can change some of the attributes of a database, including the name of the database.

You can either ALTER or RENAME the database.
- Use the AL line command to make certain changes that are supported by the ALTER DATABASE statement. DB2 Admin issues an ALTER DATABASE statement to make the changes.
- Use the ALT line command to rename a database.

Altering a database

Use ALTER to make certain changes that are supported by the ALTER DATABASE statement.

About this task

To alter a database:

Procedure

1. Enter the `al` line command against the database you want to alter, under the **Select** column on the Databases panel (ADB21D).

```
ADB21D in ------------------- DB2X Databases ----------------- Row 1 to 1 of 1
Command ===>                                              Scroll ===> CSR

Commands: GRANT  MIG  DIS  STA  STO  UTIL
Line commands:
 T - Tables  S - Table spaces  X - Indexes  G - Storage group  ICS - IC status
 DIS - Display database  STA - Start database  STO - Stop database  A - Auth
 ? - Show all line commands
                          Storage  Buffer         Created       Index
Select Name      Owner   Group    Pool     DBID  By      T E BPool    I
       *         *       *        *         * *   * *     * * *        *
------ -------- -------- -------- -------- ------ -------- - - -------- -
al     RHPDB    SMITHRJ  RHSTGRP  BP3         436 SMITHRJ    E BP0      N
```
Figure 161. Databases panel (ADB21D)

2. Alter the Buffer pool, Index Bpool, or storage group values on the Alter
   Database panel (ADB21DA) and press Enter to run ALTER DATABASE.

```
ADB21DA n ------------------- DB2X Alter Database --------------------10:02
Command ===>


 Database  . . . : RHPDB

 Buffer pool . . . BP0    (BP0-BP49, BP8K0-9, BP16K0-9, BP32K-BP32K9 or blank)

 Index Bpool . . . BP0    (BP0-BP49, BP8K0-9, BP16K0-9, BP32K-BP32K9 or blank)

 Storage group . . SYSDEFLT  >  (storage group name)
```
Figure 162. Alter Database panel (ADB21DA)

Renaming a database

Use the ALT line command to rename a database.

About this task

To rename a database with the ALT line command:

Procedure

1. In the **Select** column of the Databases panel (ADB21D), enter the ALT line
   command against the database you want to rename.

```
ADB21D in ------------------- DB2X Databases ----------------- Row 1 to 1 of 1
Command ===>                                              Scroll ===> CSR

Commands: GRANT  MIG  DIS  STA  STO  UTIL
Line commands:
 T - Tables  S - Table spaces  X - Indexes  G - Storage group  ICS - IC status
 DIS - Display database  STA - Start database  STO - Stop database  A - Auth
 ? - Show all line commands
                          Storage  Buffer         Created       Index
Select Name      Owner   Group    Pool     DBID  By      T E BPool    I
       *         *       *        *         * *   * *     * * *        *
------ -------- -------- -------- -------- ------ -------- - - -------- -
ALT    PJDB01   DSCGDB2  PJSTGRP  BP3         436 ISTJE      E BP0      N
```
Figure 163. Databases panel (ADB21D)

2. Specify a new database name on the Alter/Rename Database panel
   (ADB21DA). You can also alter the Buffer pool, Index Bpool, or storage group

values on this panel. Press Enter.

```
ADB21DA n -------------------- DB2X Alter/Rename Database -------------- 10:02
Command ===>


 New Database. . : RHPDB    Database : RHPDB

 Buffer pool . . . BP3     (BP0-BP49, BP8K0-9, BP16K0-9, BP32K-BP32K9 or blank)

 Index Bpool . . . BP0     (BP0-BP49, BP8K0-9, BP16K0-9, BP32K-BP32K9 or blank)

 Storage group . . PJSTGRP  >  (storage group name)
```

Figure 164. Alter/Rename Database panel (ADB21DA)

3. Enter NEXT on the command line on the Alter Objects panel (ADB27CA).

```
ADB27CA n ----------------------- DB2X Alter Objects --------------- Row 1 of 1
Command ===>  NEXT                                        Scroll ===> PAGE

Commands: NEXT - Generate jobs    ADD - Add objects


  ALTOPT - Change alter options
Line commands:
  A - Alter Object  D - Delete  S - Select Object  REL - Alter related
  FK - Add Foreign Key-affected tables  RI - Add RI-related tables  E - Edit DDL
  RS - Reset RI-FK flags  CX - Create index  CFK - Create foreign key

    Object   Object                                 RI RI  FK
Sel Qual     Name                Ty Info 1  Info 2  Rels Add Add Operation
    *        *                   *  *        *       *  *   *   *
--- ------->  ----------------->  -- -------- -------- ------ --- --- ---------
    DSN81010 DEPT                TB PJOBTS   PJOBTS      5 NO  NO  NONE
****************************** END OF DB2 DATA ******************************
```

Figure 165. Alter Objects panel (ADB27CA)

4. Use the ALTER Analysis Options panel (ADBP7P) to select whether you want
    to do an online analysis or a batch analysis. Batch analysis is the preferred
    method.
    *   To do a batch analysis, enter YES in the **Perform analysis in batch** field and
        press Enter.
    *   To do an online analysis, enter NO in the **Perform analysis in batch** field and
        press Enter. If the analysis process determines that SQL ALTER statements
        will accomplish the task, panel ADB27CTC is displayed, which allows you to
        select whether you want to run the SQL statements in foreground (online) or
        to generate a batch job. If you select ALTER statements, the SQL is run in the
        foreground. If you specify batch jobs, panel ADBPALT is displayed. Also, if
        the analysis process determines that a DB2 ALTER statement cannot be used,
        panel ADBPALT is displayed.

```
  ADBP7P in ----------------- DB2X ALTER Analysis Options ----------------- 14:30
  Option ===>

  Please specify the following for DB2 Admin ALTER:

  Analysis options:
    Run SQLID . . . . . . . . . . . . . .        (Blank, an SQLID, or <NONE>)
    Object Grantor  . . . . . . . . . . .        (Blank or an SQLID)
    Use DEFER YES . . . . . . . . . . . . YES     (Yes/No)
    Retain GENERATED ALWAYS:
      For ROWID . . . . . . . . . . . .          (Yes/No)
      For ROW CHANGE TIMESTAMP  . . . . .        (Yes/No)
    IDENTITY START value  . . . . . . . .        (Original, Computed)
    SEQUENCE RESTART value  . . . . . . .        (Original, Computed)
    VIEW Column List  . . . . . . . . . . YES     (Yes/No)
    Perform recovery analysis . . . . . . NO      (Yes/No)
    Enable authorization switching  . . . YES     (Yes/No)

  Perform analysis in batch . . . . . . . YES     (Yes/No)

  Show this panel prior to each use . . . YES     (Yes/No)
```

Figure 166. ALTER Analysis Options panel (ADBP7P)

5. On the ALTER - Build Analyze and Apply Job panel (ADBPALT) you can
   specify options for building the WSL or batch job that is used to implement the
   changes. After you have specified your options, press Enter to run the job.

```
ADBPALT ------------------- ALTER - Build Analyze and Apply Job ---------------
Option ===>

 Specify the following:
                                                          More:    +
 Worklist information:
   Worklist name . . . . . . .          (also used as middle qualifier in DSNs)
   Prefix for data sets  . . .

 Data set information:
   PDS final qualifiers  . . .
     Member name . . . . . . . ADBALTER
     Delete member name  . . . ADBDELET (Optional job to delete work data sets)

 Options:
   Generate online . . . . . .          (Yes/No)
   Generate one job  . . . . . YES      (Yes/No)
     Member name or prefix . . APPLY
   As work statement list  . . YES      (Yes/No)
   Content of apply job(s) . . ALL      (All, DDL)
   Unload method . . . . . . . U        (Unload, Parallel unload, HPU)
   Authorization Switch ID . .          (SQLID to sign on as, blank or NONE)
   SECADM Authorization ID . .          (An ID to sign on as, blank or NONE)
   Disable REORG optimization . YES     (Yes/No)

   Optional processes:
     Run CHECK DATA  . . . . . NO        (Yes/No)
     Run COPY  . . . . . . . . N         (after: Reload/Alter/Both/None)
     Run REORG/REBUILD . . . . N         (Mandatory, All relevant, None)
     Run RUNSTATS  . . . . . . N         (after: Reload/Alter/Both/None)
     Run REBIND  . . . . . . . NO        (Yes/No)

   Utility control options:
     Use templates . . . . . .           (Yes/No)
     Use utility options . . .           (Yes/No)

 BP - Change batch job parameters
 TU - Specify TEMPLATE usage
 UO - Customize utility options
 CO - Change options common to change functions
```

Figure 167. ALTER - Build Analyze and Apply Job panel (ADBPALT)

Changing table spaces

When you change a table space, DB2 Admin issues an ALTER TABLESPACE
statement for certain changes that are supported by the ALTER TABLESPACE
statement.

To make changes that are more complex and are not supported by the ALTER
TABLESPACE statement, DB2 Admin generates a set of batch jobs to implement
the changes.

To change a table space, you issue the AL or ALT line command.
• Use the AL line command to make changes that are supported by the ALTER
  TABLESPACE statement. Use the ALT line command to make changes supported
  by ALTER TABLESPACE and other changes not supported by ALTER
  TABLESPACE.

Alter a table space

Use the AL command to alter a table space.

About this task

To alter a table space with the AL line command:

Procedure

1. Enter the al line command against the table space you want to alter, under the **Select** column on the Table Spaces panel (ADB21S).

```
ADB21S in ------------------ DB2X Table Spaces -------------------- Row 1 of 7
Command ===>                                              Scroll ===> PAGE

Commands: GRANT  MIG  DIS  STA  STO  ALL  DROP
Line commands:
 T - Tables  D - Database  A - Auth  G - Storage group  ICS - Image copy status
 DIS - Display table space  STA - Start table space  STO - Stop table space
 ? - Show all line commands

Select Name     DB Name   Parts Bpool  L E S I C Tables  Act. pages  Segsz T L
       *        *             * *       * * * * *   *           *      * * *
------ -------- -------- ------ ------ - - - - - ------ ----------- ------ - -
al     DEPT     DBN00793     1 BP0    R N T N Y     0         -1     32 G Y
       DEPT08TS DEPT0818     1 BP0    R N T N Y     0         -1     32 G Y
       DEPT     DSN00818     1 BP0    R N A Y Y     1         -1     32 G Y
       DEPT     DSN008RN     1 BP0    R N T N Y     0         -1     32 G Y
       DEPT     DSN008XX     1 BP0    R N T N Y     0         -1     32 G Y
       DEPTTS   DSNDB04      2 BP0    R N T N Y     0         -1     32 R Y
       DEPTTS   DSNRR        2 BP0    R N T N Y     0         -1     32 R Y
****************************** END OF DB2 DATA ******************************
```

Figure 168. Table Spaces panel (ADB21S)

2. Alter the table space attributes or one or more partitions within a table space. The SQL ALTER TABLESPACE statement is performed when you change a parameter and press Enter. Changes to other parameters, such as the Primary Quantity, do not take effect until the object is reorganized.

```
ADB21SA n -------------------- DB2X Alter Table Space ------------- Row 1 of 1
Command ===>                                              Scroll ===> PAGE

Line commands:
 D - Display Database   I - Interpret

ALTER TABLESPACE : DBN00793.DEPT          (Partition by Growth      )

Buffer Pool  . . . . BP0      Close Rule  . . . YES Max Rows  . . 255
Lock Size  . . . . . ROW      Lock Part . . . . NO  Lock Max  . . SYSTEM
Max Partitions . . . 256      LOG . . . . . . . YES DSSIZE  . . . 4 G
SEGSIZE  . . . . . . 32       MEMBER CLUSTER  . NO

        Primary    Secondary  Free Pct Com E T S
S   Part Quantity  Quantity   Page Free prs R M T VCAT     Stogroup GBPCache
- ------ ---------- ---------- ---- ---- --- - - - -------- -------> -------
All Part
      1          -1         -1    0    5 NO    Y I DSNA     SYSDEFLT CHANGED
****************************** END OF DB2 DATA ******************************
```

Figure 169. Alter Table Space panel (ADB21SA)

Results

For partitioned table spaces, a detail line is displayed for each partition. You can alter any partition by updating the attributes, such as **Pct Free**. To apply the same change to all partitions within the table space, provide a value on the **All Part** field.

To change certain parameters, you must stop and restart the associated object. In these cases, DB2 Admin runs a STOP table space or STOP index (or partition) command and checks that the object is in a fully-stopped state. If stopped, it runs an ALTER TABLESPACE statement, followed by a START command. If the object is not in a fully-stopped state, the STOP Check – Action panel, shown in the following figure, prompts you to perform one of the following actions:

- Check again and continue if in STOP state.
- Issue the ALTER statement.
- Cancel the operation.

If an object is not stopped when the ALTER TABLESPACE statement runs (for example, if others are holding locks on the object), a -626 SQLCODE is displayed.

```
DB2 ADMIN ------------------- DB2X STOP Check - Action ----- Row 1 to 11 of 15
Option ===>                                              Scroll ===> PAGE

Object is not in a fully-stopped state (STATUS field has STOP), and must be in
order for the pending actions to be successful.  The current USE information is
displayed below.
What do you want to do now:
1 - Re-check and continue if in STOP state. Re-display USE if not
2 - Perform any pending actions, regardless of the object's state
3 - Exit and do not perform any pending actions

*******************************************************************************
DSNT360I  @ **********************************
DSNT361I  @ *  DISPLAY DATABASE SUMMARY
          *     GLOBAL USE
DSNT360I  @ **********************************
DSNT362I  @     DATABASE = DSN8D81A  STATUS = RW
                DBD LENGTH = 16142
DSNT397I  @
NAME    TYPE PART STATUS             CONNID   CORRID       USERID
-------- ---- ---- ------------------ -------- ------------ --------
DSN8S81D TS        STOPP              TSO      SYSADM       SYSADM
    -              MEMBER NAME V81A
******* DISPLAY OF DATABASE DSN8D81A ENDED     **********************
DSN9022I  @ DSNTDDIS 'DISPLAY DATABASE' NORMAL COMPLETION
***************************** Bottom of data ********************************
```

Figure 170. STOP Check — Action (ADBWSTOP)

Examples of altering and redefining a table space

Use the AL command to alter a table space and the ALT command to redefine a table space. In these examples, an ALTER TABLESPACE statement changes the table space. The table space is not dropped and recreated.

Reducing the MAXPARTITIONS value for a Partition-by-growth (PBG) table space by altering the table space:
About this task

You can use the AL command to reduce the MAXPARTITIONS value for a PGB table space.

To reduce the MAXPARTITIONS value:

Procedure

1. From the Table Spaces panel (ADB21S), issue the AL line command against the table space that you want to reduce the MAXPARTITIONS value for. The Alter Table Space panel (ADB21SAR) is displayed.

2. On the Alter Table Space panel, type a new value in the **Max Partitions** field and press Enter.

    An ALTER TABLESPACE statement is executed and the MAXPARTITIONS value is reduced.

Reducing the MAXPARTITIONS value for a Partition-by-growth (PBG) table space by redefining the table space:
About this task

You can use the ALT command to reduce the `MAXPARTITIONS` value for a PGB table space.

To reduce the MAXPARTITIONS value:

Procedure
1. From the Table Spaces panel (ADB21S), issue the ALT line command against the table space that you want to reduce the MAXPARTITIONS value for. The Alter Table Space panel (ADB21SAR) is displayed.
2. On the Redefine Table Space panel, type a new value in the **Max Partitions** field, and type `Continue` on the command line. Press Enter.

    The Alter Objects panel panel (ADB27CA) is displayed.
3. On the Alter Objects panel, type `ALTER` on the command line. Press Enter.
4. On the ALTER Analysis Options panel (ADBP7P) select an online analysis or a batch analysis.
5. On the ALTER - Build Analyze and Apply Job panel (ADBPALT), specify options for building the WSL or batch job that is used to implement the changes. After you specify your options, press Enter to run the job.

Redefining a non-partitioned table space

Use the ALT line command to redefine a table space.

About this task

To redefine a table space with the ALT line command:

Procedure
1. In the Select column of the Table Spaces panel (ADB21S), enter the `alt` line command against the table space you want to redefine.

```
ADB21S in ------------------ DB2X Table Spaces --------------- Row 1 to 5 of 5
Command ===>                                                  Scroll ===> CSR

Commands: GRANT  MIG  DIS  STA  STO  ALL
Line commands:
 T - Tables  D - Database  A - Auth  G - Storage group  ICS - Image copy status
 DIS - Display table space  STA - Start table space  STO - Stop table space
 ? - Show all line commands

Select Name      DB Name   Parts Bpool  L E S I C Tables  Act. pages  Segsz T L
       *         *          * *         * * * * *    *         *        * * *
------ -------- -------- ------ ------ - - - - - ------ ----------- ------ - -
alt    TSFGR    DBFGR        0 BP0    A N A N Y      1           0      4   Y
       TSFGRPBR DBFGR        3 BP0    A N C N Y      1           0      4 R Y
       TSFGRRO1 DBFGRRO1     3 BP0    A N A N Y      1           0     64 R Y
       TSFGRRO2 DBFGRRO2     3 BP0    A N A N Y      1           0     64 R Y
       TSFGRROO DBFGRROO     2 BP0    A N T N Y      0           0     64 R Y
```

Figure 171. Table Spaces panel (ADB21S)

2. Change the parameters to redefine the table space and then enter `continue` on the command line on the Redefine Table Space panel (ADB21SAR).

```
ADB21SAR  ------------------ DB2X Redefine Table Space ------- Row 1 to 1 of 1
Command ===> next                                             Scroll ===> CSR

Commands: NEXT  ORIGINAL  MAKEPBG  MAKEPBR
Line commands: S - Split part   R - Remove part  O - Original data
               C - Clear data
CREATE TABLESPACE: TSFGR    IN  DBFGR

Numparts  . . . . . 0          Large . . . . . NO    LOB . . . . . NO
Define  . . . . . . YES        DSSIZE  . . . .       LOG . . . . . YES
Member Cluster  . . NO         SEGSIZE . . . . 4     CCSID . . . . EBCDIC
Buffer Pool . . . . BP0        Close Rule  . . YES   Max Rows  . . 255
Lock Size . . . . . ANY        Lock Part . . . NO    Lock Max  . . SYSTEM
Max Partitions  . . 0
                               Free  Pct     E T S
S    Part       Pqty     Sqty Page Free Compr R M T VCAT    Stogroup GBPCache
- ------ ---------- ---------- ---- ---- ----- - - - -------- -------> --------

      0         12         12    1    4 YES   N Y I DSNA    SYSDEFLT CHANGED
```

Figure 172. Redefine Table Space panel (ADB21SAR)

If you are converting a segmented table space to a partitioned table space, the Alter tablespace - Partitioning methods panel is displayed, as shown in the following figure:

```
ADB2CONF  -- DB2X Alter tablespace - Partitioning methods --------- 19:28

 Please choose partitioning method for the table space to be altered.




 Select a choice
   1.  Use table-controlled partitioning (recommended)
   2.  Use index-controlled partitioning
```

Figure 173. Alter tablespace - Partitioning methods panel (ADB2CONF)

Select option 1 to use table-controlled partitioning. When the Alter Table panel (ADB21TAP) is displayed, specify the partitioning key for defining the table partitions. If the ALT - Index-controlled Partitioning panel (ADB21XAP) is displayed, you can re-define an existing non-partitioning index to a partitioning

index. If the Create Partitioning Index panel (ADB21SAX) is displayed you can create a partitioning index. It is recommended that you use table-controlled partitioning because it will eventually replace index-controlled partitioning.

3. Enter NEXT on the command line on the Alter Objects panel (ADB27CA).

```
ADB27CA n ---------------------- DB2X Alter Objects --------------- Row 1 of 1
Command ===>  NEXT                                           Scroll ===> PAGE

Commands: NEXT - Generate jobs   ADD - Add objects


   ALTOPT - Change alter options
Line commands:
  A - Alter Object  D - Delete  S - Select Object  REL - Alter related
  FK - Add Foreign Key-affected tables  RI - Add RI-related tables  E - Edit DDL
  RS - Reset RI-FK flags  CX - Create index  CFK - Create foreign key

    Object   Object                                 RI RI  FK
Sel Qual     Name               Ty Info 1   Info 2    Rels Add Add Operation
    *        *                  * *         *         * *  *   *
--- -------> ----------------->  -- -------- -------- ------ --- --- ---------
    DSN81010 DEPT               TB PJOBTS   PJOBTS       5 NO  NO  NONE
***************************** END OF DB2 DATA ******************************
```

Figure 174. Alter Objects panel (ADB27CA)

4. Use the ALTER Analysis Options panel (ADBP7P) to select whether you want to do an online analysis or a batch analysis. Batch analysis is the preferred method.

   - To do a batch analysis, enter YES in the **Perform analysis in batch** field and press Enter.
   - To do an online analysis, enter NO in the **Perform analysis in batch** field and press Enter. If the analysis process determines that SQL ALTER statements will accomplish the task, panel ADB27CTC is displayed, which allows you to select whether you want to run the SQL statements in foreground (online) or to generate a batch job. If you select ALTER statements, the SQL is run in the foreground. If you specify batch jobs, panel ADBPALT is displayed. Also, if the analysis process determines that a DB2 ALTER statement cannot be used, panel ADBPALT is displayed.

```
ADBP7P in ----------------- DB2X ALTER Analysis Options ----------------- 14:30
Option ===>

Please specify the following for DB2 Admin ALTER:

Analysis options:
  Run SQLID . . . . . . . . . . . . .          (Blank, an SQLID, or <NONE>)
  Object Grantor  . . . . . . . . . . .        (Blank or an SQLID)
  Use DEFER YES . . . . . . . . . . . . YES    (Yes/No)
  Retain GENERATED ALWAYS:
    For ROWID . . . . . . . . . . . .           (Yes/No)
    For ROW CHANGE TIMESTAMP  . . . . .          (Yes/No)
  IDENTITY START value  . . . . . . . .        (Original, Computed)
  SEQUENCE RESTART value  . . . . . . .        (Original, Computed)
  VIEW Column List  . . . . . . . . . . YES    (Yes/No)
  Perform recovery analysis . . . . . . NO     (Yes/No)
  Enable authorization switching  . . . YES    (Yes/No)

Perform analysis in batch . . . . . . . YES    (Yes/No)

Show this panel prior to each use . . . YES    (Yes/No)
```

Figure 175. ALTER Analysis Options panel (ADBP7P)

5. On the ALTER - Build Analyze and Apply Job panel (ADBPALT) you can specify options for building the WSL or batch job that is used to implement the changes. After you have specified your options, press Enter to run the job.

```
ADBPALT ------------------ ALTER - Build Analyze and Apply Job ---------------
Option ===>

 Specify the following:
                                                            More:    +
 Worklist information:
   Worklist name . . . . . . .           (also used as middle qualifier in DSNs)
   Prefix for data sets  . . .

 Data set information:
   PDS final qualifiers  . . .
     Member name . . . . . . . ADBALTER
     Delete member name  . . . ADBDELET (Optional job to delete work data sets)

 Options:
   Generate online . . . . . .           (Yes/No)
   Generate one job  . . . . . YES       (Yes/No)
     Member name or prefix . . APPLY
   As work statement list  . . YES       (Yes/No)
   Content of apply job(s) . . ALL       (All, DDL)
   Unload method . . . . . . . U         (Unload, Parallel unload, HPU)
   Authorization Switch ID . .           (SQLID to sign on as, blank or NONE)
   SECADM Authorization ID . .           (An ID to sign on as, blank or NONE)
   Disable REORG optimization . YES       (Yes/No)

   Optional processes:
     Run CHECK DATA  . . . . . NO        (Yes/No)
     Run COPY  . . . . . . . . N         (after: Reload/Alter/Both/None)
     Run REORG/REBUILD . . . . N         (Mandatory, All relevant, None)
     Run RUNSTATS  . . . . . . N         (after: Reload/Alter/Both/None)
     Run REBIND  . . . . . . . NO        (Yes/No)

   Utility control options:
     Use templates . . . . . .           (Yes/No)
     Use utility options . . .           (Yes/No)

 BP - Change batch job parameters
 TU - Specify TEMPLATE usage
 UO - Customize utility options
 CO - Change options common to change functions
```

Figure 176. ALTER - Build Analyze and Apply Job panel (ADBPALT)

Redefining an existing partitioned table space (table-controlled partitioning)

Use the ALT line command to redefine a table space.

About this task

To redefine an existing partitioned table space using table-controlled partitioning:

Procedure

1. In the Select column of the Table Spaces panel (ADB21S), enter the ALT line command against the table space you want to redefine.

```
ADB21S in ----------------- DSNB Table Spaces --------------- Row 1 to 4 of 4
Command ===>                                              Scroll ===> CSR

Commands: GRANT  MIG  DIS  STA  STO  ALL  CT  DROP
Line commands:
 T - Tables  D - Database  A - Auth  G - Storage group  ICS - Image copy status
 DIS - Display table space  STA - Start table space  STO - Stop table space
 ? - Show all line commands

Select Name      DB Name    Parts Bpool  L E S I C Tables  Act. pages  Segsz T L
       *         *            * *        * * * * *    *          *       * * *
------ -------- -------- ------ ------ - - - - - ------ ----------- ------ - -
ALT    TSFGRTB  DBFGRTB      4 BP0    A N A N Y    1          -1     32 R Y
       TSFGRTB1 DBFGRTB1     3 BP0    A N A N Y    1          -1     32 R Y
       TSFGRTB2 DBFGRTB2     5 BP0    P N A N N    1          -1      0   Y
       TSFGRTB4 DBFGRTB4     5 BP0    P N A N N    1          -1     32 R Y
```

Figure 177. Table Spaces panel (ADB21S)

2. On the Change Management Prompt panel, enter No.

```
ADB27C0 n ------------------------- DSNB Alter -------------------------------
C EsssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssN
  e ADB2CMRO  ----------- DSNB Change Management Prompt ----------- 00:52 e
  e                                                                         e
  e Change Management is optional for SQLID:  RIVERAF                       e
  e                                                                         e
  e Do you wish to use Change Management for this function: N   (Yes/No)  e
  e                                                                         e
  e                                                                         e
  e                                                                         e
  DsssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssM
```

Figure 178. Change Management Prompt panel (ADB27C0)

3. On the Redefine Table Space panel, enter VALUES on the command line.
4. On the Alter Partitioned Table panel (ADB21TAV), edit the LIMITKEY value
   that you want to update and then enter NEXT on the command line.

```
ADB21TAV  --------------- DSNB Alter Partitioned Table ------- Row 1 to 4 of 4
Command ===>                                              Scroll ===> CSR

Commands: NEXT  COLUMNS


ALTER TABLE : RIVERAF.TBFGRTB


Sel   Part Limit Key Value
--- ------ ------------------------------------------------------------------->
       1 '1111       ',1111.
       2 '2224       ',2224.
       3 '3333       ',3333.
       4 '6666       ',6666.
```

Figure 179. Alter Partitioned Table panel (ADB21TAV)

5. On the Redefine Table Space panel, enter NEXT on the command line.
6. On the Alter Objects panel (ADB27CA), enter NEXT on the command line.

```
ADB27CA n --------------------- DSNB Alter Objects ---------- Row 1 to 1 of 1
Command ===>                                                 Scroll ===> CSR

Commands: NEXT - Generate jobs  ADD - Add objects
   ALTOPT - Change alter options
Line commands:
 A - Alter object  D - Delete  S - Select object  REL - Alter related
 FK - Add FK-affected tables  RI - Add RI-related tables  E - Edit view DDL
 RS - Reset RI-FK flags  CX - Create index  CFK - Create foreign key

    Object   Object                                RI RI FK
Sel Qual     Name             Ty Info 1  Info 2   Rels Add Add Operation
    *        *                * *        *         * *   *   *
--- -------> -----------------> -- -------> -------> ------ --- --- -----------
    DBFGRTB  TSFGRTB          TS                       NA  NA  MODIFY
```

Figure 180. Alter Objects panel (ADB27CA)

7. On the ALTER - Build Analyze and Apply Job panel (ADBPALT) you can
   specify options for building the WSL or batch job that is used to implement the
   changes. After you have specified your options, press Enter to run the job.

```
ADBPALT ------------------- ALTER - Build Analyze and Apply Job ---------------
Command ===>

 Specify the following:

 Worklist information:
   Worklist name . . . . . . . TESTA     (also used as middle qualifier in DSNs)
   Prefix for data sets  . . . RIVERAF

 Data set information:
   PDS final qualifiers  . . . TESTA.JCL
     Member name . . . . . . . ADBALTER
     Delete member name  . . . ADBDELET (Optional job to delete work data sets)

 Options:
   Generate online . . . . . . NO        (Yes/No)
   Generate one job  . . . . . YES       (Yes/No)
     Member name or prefix . . APPLY
   As work statement list  . . NO        (Yes/No)
   Content of apply job(s) . . ALL       (All, DDL)
   Unload method . . . . . . . U         (Unload, Parallel unload, HPU)
   Authorization Switch ID . . <NONE>    (SQLID to sign on as, blank or <NONE>)
   SECADM Authorization ID . .           (An ID to sign on as, blank or <NONE>)
   Disable REORG optimization . YES       (Yes/No)

   Optional processes:
     Run CHECK DATA  . . . . . NO        (Yes/No)
     Run COPY  . . . . . . . . N         (after: Reload/Alter/Both/None)
     Run REORG/REBUILD . . . . N         (Mandatory, All relevant, None)
     Run RUNSTATS  . . . . . . N         (after: Reload/Alter/Both/None)
     Run REBIND  . . . . . . . N         (Mandatory, All relevant, None)

   Utility control options:
     Use templates . . . . . . NO        (Yes/No)
     Use utility options . . . NO        (Yes/No)

 BP - Change batch job parameters
 TU - Specify TEMPLATE usage
 UO - Customize utility options
 CO - Change options common to change functions
```

Figure 181. ALTER - Build Analyze and Apply Job panel (ADBPALT)

8. On the Apply Job Data Set panel (ADBPALTJ), enter your data set name, then
   press Enter.

```
ADBPALTJ ------ Alter - Apply Job Data Set --------------------- 01:21

Enter/verify the following:
 Data Set Name . . . RIVERAF.TESTA.APPLYJCL
```

Figure 182. Alter - Apply Job Data Set (ADBPALTJ)

Redefining an existing partitioned table space (index-controlled partitioning)

Use the ALT line command to redefine a table space.

About this task

To redefine an existing partitioned table space using index-controlled partitioning:

Procedure

1. In the Select column of the Table Spaces panel (ADB21S), enter the ALT line command against the table space you want to redefine.

```
ADB21S in ----------------- DSNB Table Spaces --------------- Row 1 to 5 of 5
Command ===>                                              Scroll ===> CSR

Commands: GRANT  MIG  DIS  STA  STO  ALL  CT  DROP
Line commands:
 T - Tables  D - Database  A - Auth  G - Storage group   ICS - Image copy status
 DIS - Display table space  STA - Start table space  STO - Stop table space
 ? - Show all line commands

Select Name       DB Name    Parts Bpool  L E S I C Tables  Act. pages  Segsz T L
       *          *            * *        * * * * *    *           *      * * *
------ -------- -------- ------ ------ - - - - - ------ ----------- ------ - -
ALT    TSFGRIX  DBFGRIX      3 BP0    A N A N Y    1          -1      0   Y
       TSFGRIX1 DBFGRIX1     3 BP0    A N A N Y    1          -1      0   Y
       TSFGRIX2 DBFGRIX2     4 BP0    P N A N N    1          -1      0   Y
       TSFGRIX3 DBFGRIX3     6 BP1    A N A N N    1          -1      0   Y
       TSFGRIXR DBFGRIXR     3 BP0    A N A N Y    1          -1      0   Y
```

Figure 183. Table Spaces panel (ADB21S)

2. On the Change Management Prompt panel, enter No.

```
ADB27C0 n ------------------------ DSNB Alter -----------------------------
C EssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssN
  e ADB2CMRO  ----------- DSNB Change Management Prompt ----------- 00:52 e
  e                                                                       e
  e Change Management is optional for SQLID:  RIVERAF                     e
  e                                                                       e
  e Do you wish to use Change Management for this function: N   (Yes/No)  e
  e                                                                       e
  e                                                                       e
  e                                                                       e
  DssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssM
```

Figure 184. Change Management Prompt panel (ADB27C0)

3. On the Redefine Table Space panel (ADB21SAR), increase the Numparts value and press Enter. For example, change the Numparts value from 3 to 4. After you press Enter, you should see a new partition row added to the list of partitions.

4. On the Redefine Table Space panel, enter NEXT on the command line.

```
ADB21SAR  ------------------ DSNB Redefine Table Space ------- Row 1 to 3 of 3
Command ===>                                               Scroll ===> CSR

Commands: NEXT  ORIGINAL  BALANCE  VALUES  MAKEPBG  MAKEPBR
Line commands: S - Split part   R - Remove part  O - Original data
              C - Clear data
CREATE TABLESPACE: TSFGRIX  IN  DBFGRIX


Numparts  . . . . . 4                               LOB . . . . . NO
Define  . . . . . . YES         DSSIZE  . . . .      LOG . . . . . YES
Member Cluster  . . NO          SEGSIZE . . . . 0    CCSID . . . . EBCDIC
Buffer Pool . . . . BP0         Close Rule  . . YES  Max Rows  . . 255
Lock Size . . . . . ANY         Lock Part . . . NO   Lock Max  . . SYSTEM
Max Partitions  . . 0
                                  Free  Pct     E T S
S    Part       Pqty       Sqty Page Free Compr R M T VCAT    Stogroup GBPCache
- ------ ---------- ---------- ---- ---- ----- - - - -------- -------> --------
Default:        12         -1    0    5 NO    N Y I DSNB    SYSDEFLT CHANGED
      1
      2
      3
```

Figure 185. Redefine Table Space panel (ADB21SAR)

> **Note:** When redefining an existing table space with index-controlled
> partitioning, you can view LIMITKEY values for each partition by entering
> VALUES on the command line of the Redefine Table Space panel (ADB21SAR).
> You can edit exiting LIMITKEY values by increasing the Numparts value on
> ADB21SAR and entering NEXT on the command line. If you do not increase the
> Numparts value and only want to update the existing LIMITKEY values, you
> should navigate to the Indexes panel and use the ALT function on the associated
> index.

5. On the Redefine Partitioning Index panel (ADB21SAX), enter NEXT on the
   command line.

```
ADB21SAX  -------------------- DSNB Redefine Partitioning Index Row 1 to 4 of 4
Command ===>                                               Scroll ===> CSR

Commands: NEXT  ORIGINAL  BALANCE  VALUES

CREATE INDEX  RIVERAF  > . IXFGRIX            >
        ON  RIVERAF     . TBFGRIX

Unique        ===>          Where Not Null  ===>     Cluster     ===> /
Buffer pool   ===> BP0      Close rule      ===> YES Copy Allowed ===> NO
Piecesize     ===>          Define          ===>     Defer       ===>
                            Padded          ===> NO
( Column List ) ===> F1,F3




        Primary    Secondary  Free Pct      S
S    Part Quantity   Quantity  Page Free Erase T VCAT    Stogroup GBPCache
- ------ ---------- ---------- ---- ---- ----- - -------- -------> --------
Default        12         -1    0   10       I DSNB    SYSDEFLT CHANGED
      1        12         -1    0   10       I DSNB    SYSDEFLT CHANGED
      2        12         -1    0   10       I DSNB    SYSDEFLT CHANGED
      3        12         -1    0   10       I DSNB    SYSDEFLT CHANGED
      4
```

Figure 186. Redefine Partitioning Index panel (ADB21SAX)

6. On the Limit Key Values panel (ADB21SAV), enter a LIMITKEY value for the added partition and then enter NEXT on the command line.

```
ADB21SAV  -------------------- DSNB Limit Key Values --------- Row 1 to 4 of 4
Command ===>                                              Scroll ===> CSR
LIMITKEY values required
Commands: NEXT    COLUMNS
Index columns:  F1,F3




Sel   Part Limit Key Value
--- ------ ------------------------------------------------------------------->
       1 '1111          ',1111.
       2 '3333          ',3333.
       3 '5555          ',5555.
       4 '6666          ',6666.
```

Figure 187. Limit Key Values panel (ADB21SAV)

7. On the Alter Objects panel (ADB27CA), enter NEXT on the command line.

```
ADB27CA n --------------------- DSNB Alter Objects ---------- Row 1 to 1 of 1
Command ===>                                              Scroll ===> CSR

Commands: NEXT - Generate jobs  ADD - Add objects
   ALTOPT - Change alter options
Line commands:
 A - Alter object  D - Delete  S - Select object  REL - Alter related
 FK - Add FK-affected tables  RI - Add RI-related tables  E - Edit view DDL
 RS - Reset RI-FK flags  CX - Create index  CFK - Create foreign key

    Object  Object                                 RI RI  FK
Sel Qual    Name             Ty Info 1   Info 2    Rels Add Add Operation
    *       *                *  *         *           * *  *   *
--- ------> -----------------> -- ------> -------> ------ --- --- -----------
    DBFGRIX TSFGRIX          TS                      NA  NA  MODIFY
```

Figure 188. Alter Objects panel (ADB27CA)

8. On the ALTER - Build Analyze and Apply Job panel (ADBPALT) you can specify options for building the WSL or batch job that is used to implement the changes. After you have specified your options, press Enter to run the job.

```
ADBPALT ------------------ ALTER - Build Analyze and Apply Job ---------------
Command ===>

 Specify the following:

 Worklist information:
   Worklist name . . . . . . . TESTI    (also used as middle qualifier in DSNs)
   Prefix for data sets  . . . RIVERAF

 Data set information:
   PDS final qualifiers  . . . TESTI.JCL
     Member name . . . . . . . ADBALTER
     Delete member name  . . . ADBDELET (Optional job to delete work data sets)

 Options:
   Generate online . . . . . . NO       (Yes/No)
   Generate one job  . . . . . YES      (Yes/No)
     Member name or prefix . . APPLY
   As work statement list  . . NO       (Yes/No)
   Content of apply job(s) . . ALL      (All, DDL)
   Unload method . . . . . . . U        (Unload, Parallel unload, HPU)
   Authorization Switch ID . . <NONE>   (SQLID to sign on as, blank or <NONE>)
   SECADM Authorization ID . .          (An ID to sign on as, blank or <NONE>)
   Disable REORG optimization . YES      (Yes/No)

   Optional processes:
     Run CHECK DATA  . . . . . NO       (Yes/No)
     Run COPY  . . . . . . . . N        (after: Reload/Alter/Both/None)
     Run REORG/REBUILD . . . . N        (Mandatory, All relevant, None)
     Run RUNSTATS  . . . . . . N        (after: Reload/Alter/Both/None)
     Run REBIND  . . . . . . . N        (Mandatory, All relevant, None)

   Utility control options:
     Use templates . . . . . . NO       (Yes/No)
     Use utility options . . . NO       (Yes/No)

 BP - Change batch job parameters
 TU - Specify TEMPLATE usage
 UO - Customize utility options
 CO - Change options common to change functions
```

Figure 189. ALTER - Build Analyze and Apply Job panel (ADBPALT)

9. On the Apply Job Data Set panel (ADBPALTJ), enter your data set name, then press Enter.

```
ADBPALTJ ------ Alter - Apply Job Data Set ---------------------- 01:21

Enter/verify the following:
 Data Set Name . . . RIVERAF.TESTA.APPLYJCL
```

Figure 190. Alter - Apply Job Data Set (ADBPALTJ)

Changing tables

With DB2 Admin, you can make changes to a table and its attributes.

DB2 Admin enables you to perform the following tasks:
- Change the database, table space, owner, and the name of a table
- Modify the definitions of table columns (with some restrictions)
- Change the sequence of the columns in a table
- Drop columns
- Insert new columns
- Drop and add unique, check, and foreign key constraints

- Modify table attributes such as auditing, data capture, validation procedure, restrict on drop, index access, and append processing.
- Modify the table's data organization
- Activate and deactivate row and column access control
- Drop and add column masks
- Add system or business-time periods
- Drop and add versioning
- Add or alter partitions
- Add partitioning keys
- Drop and add clone tables

Restrictions:
- Changes to column names are retrofitted into views. All other column actions are not retrofitted, and any changes to a column's data type are not verified against the views.
- All columns comprising the partitioning columns of the table cannot be dropped.
- A warning is displayed if you attempt to modify columns in the primary key. With line command UP (update primary key), you can circumvent the warning. You can use the ADDFK primary command to propagate the primary key update to foreign-key related tables.
- If you modify columns that are in a foreign key, DB2 Admin does not automatically modify the primary key of parent tables. To propagate the column updates to primary and foreign key tables, use the ADD primary command from the Alter Table panel (ADB27C) to initiate the Alter Tables dialog, where RI-related tables or other tables can be included in the Alter JCL stream.
- DB2 Admin informs you when a specific data type conversion is allowed. See Chapter 30, "DB2 Admin data type conversions," on page 1053.
- If you modify a table that has a security label column, you cannot specify the value for HPU in the **Unload Method** field on the Alter Parameters panel.
- On the Alter Parameters panel, you cannot specify the value HPU in the **Unload Method** field if you are creating a work statement list. For work statement lists, you can choose the Unload value.
- HPU cannot be used when altering a table with LOB columns.
- The HPU PARMLIB parameter must be set to the default value.

Altering or redefining a table with the ALT command

Use the ALT line command to alter or redefine a table.

About this task

To alter or redefine a table with the ALT line command:

Procedure

1. In the **Sel** column of the Tables, Views, and Aliases panel (ADB21T), enter the ALT line command against the table that you want to alter or redefine.

```
ADB21T in --------------- DB2X Tables, Views, and Aliases --------- Row 1 of 1
Command ===>                                               Scroll ===> PAGE

Commands: GRANT  MIG  ALL
Line commands:
 C - Columns  A - Auth  L - List  X - Indexes  S - Table space  D - Database
 V - Views  T - Tables  P - Plans  Y - Synonyms  SEL - Select prototyping
 ? - Show all line commands

Sel   Name             Schema   T DB Name  TS Name   Cols       Rows Chks C
      *                *        * *        *         *             *    * *
----- ---------------- -------- - -------- -------- ------ ----------- ---- -
ALT   DEPT             DSN11010 T DSN8D10A DSN8S10D      5          14    0
****************************** END OF DB2 DATA ******************************
```

Figure 191. Tables, Views, and Aliases panel (ADB21T)

2. On the ALTER Table panel (ADB27C), change any attributes of the table. In this
   example, the **New Schema** and the **New name** are changed. Enter NEXT on the
   command line.

```
ADB27C in ------------------ DB2X ALTER Table --------------- Row 1 to 5 of 5
Command ===>  NEXT                                          Scroll ===> CSR

New schema . . BDB  >                     Old schema: DSN11010
New name . . . BDBCATVT         >         Old name  : DEPT
Partitions . : 1                          New DB  . . DSN8D10A
Rows per page: 47                         New TS  . . DSN8S10D

Commands:  NEXT  CONSTRAINTS  TBLOPTS        HASH
Line commands:
 I - Insert  U - Update  D - Delete  R - Repeat  LAB - Label  COM - Comment
 M - Move    A - After   B - Before  X - Index   RES - Reset update
UM - Update XML modifiers
                                                         Old Operation
Sel Column Name       Col No Col Type    Length  Scale N D Col No Type
    *                     * *            *       * * *      * *
--- ----------------->  ------ -------- ----------- ------ - - ------ ---------

    DEPTNO                 1 CHAR             3       0 N N    1
    DEPTNAME               2 VARCHAR         36       0 N N    2
    MGRNO                  3 CHAR            16       0 Y Y    3
    ADMRDEPT               4 CHAR             3       0 N N    4
    LOCATION               5 CHAR            16       0 Y Y    5
****************************** END OF DB2 DATA ******************************
```

Figure 192. ALTER Table panel (ADB27C)

3. Optional: To make additional changes to the table, such as specifying a period
   definition for the table, enter TBLOPTS on the command line. After making the
   additional changes, enter CONTINUE on the command line.

   a. On the Alter - Table Options panel (ADBP7TOP), make additional changes
      to the table.

```
ADBP7TOP in ------------ DB2X ALTER - Table Options----------- Row 1 to 5 of 5
Command ===>

New schema . . BDB  >
New name . . . BDBCATVT          >

Enter table options below:

AUDIT  . . . . . . . . .          (None, Changes, or All)
DATA CAPTURE . . . . . .          (None/Changes)
VALIDPROC  . . . . . . .          (NULL/Program name)
RESTRICT ON DROP . . . .          (Yes/No)
VOLATILE . . . . . . . .          (Yes/No)
APPEND . . . . . . . . .
LABEL  . . . . . . . .
COMMENT  . . . . . . . .
Business period  . . . .          (Yes/No)
  Begin column . . . . .          ?        > (? to lookup)
  End column . . . . . .                   > (? to lookup)
System period  . . . . .          (Yes/No)
Versioning  . . . . . . .         (Yes, No, or Chg)
ENABLE ARCHIVE  . . . . .         (Yes, No, or Chg)
***************************** END OF DB2 DATA *****************************
```

Figure 193. Alter - Table Options panel (ADBP7TOP)

    b. Enter + to specify Begin and End column names for the business period on the panel that appears (ADBP7TOC).

```
ADBP7TOC DTEST - DSNA BUSINESS_TIME begin column ------------- Row 1 to 6 of 6
Command ===>                                              Scroll ===> CSR
Select by typing '+'
New schema . : BDB  >
New name . . : BDBCATVT
                                                         Old Operation
Sel Column Name        Col No Col Type    Length  Scale N D Col No Type
    *                       *  *               *       * * *     * *
--- ------------------ ------ -------- ----------- ------ - - ------ ---------
    C1                      1 CHAR            10      0 N N     1
    I1                      2 INTEGER          4      0 N N     2
    I2                      3 INTEGER          4      0 Y Y     3
    SYS_START               4 TIMESTMP        13     12 N Q     4
    SYS_END                 5 TIMESTMP        13     12 N R     5
    TC                      6 TIMESTMP        10      6 N Y     6
***************************** END OF DB2 DATA *****************************
```

4. Enter NEXT on the command line of the ALTER Objects panel (ADB27CA).

```
ADB27CA n ---------------------- DB2X Alter Objects --------------- Row 1 of 1
Command ===>  NEXT                                         Scroll ===> PAGE

Commands: NEXT - Generate jobs   ADD - Add objects


  ALTOPT - Change alter options
Line commands:
  A - Alter Object  D - Delete  S - Select Object  REL - Alter related
  FK - Add Foreign Key-affected tables  RI - Add RI-related tables  E - Edit DDL
  RS - Reset RI-FK flags  CX - Create index  CFK - Create foreign key

    Object   Object                                  RI RI  FK
Sel Qual     Name             Ty Info 1   Info 2    Rels Add Add Operation
    *        *                *  *         *          *  *   *   *
--- -------> ------------------> -- -------- -------- ------ --- --- ---------
    DSN81010 DEPT             TB PJOBTS   PJOBTS       5 NO  NO  NONE
***************************** END OF DB2 DATA ******************************
```

Figure 194. Alter Objects panel (ADB27CA)

5. Use the ALTER Analysis Options panel (ADBP7P) to select whether you want to do an online analysis or a batch analysis. Batch analysis is the preferred method.
   - To do a batch analysis, enter YES in the **Perform analysis in batch** field and press Enter.
   - To do an online analysis, enter NO in the **Perform analysis in batch** field and press Enter. If the analysis process determines that SQL ALTER statements will accomplish the task, panel ADB27CTC is displayed, which allows you to select whether you want to run the SQL statements in foreground (online) or to generate a batch job. If you select ALTER statements, the SQL is run in the foreground. If you specify batch jobs, panel ADBPALT is displayed. Also, if the analysis process determines that a DB2 ALTER statement cannot be used, panel ADBPALT is displayed.

```
ADBP7P in ----------------- DB2X ALTER Analysis Options ----------------- 14:30
Option ===>

Please specify the following for DB2 Admin ALTER:

Analysis options:
  Run SQLID . . . . . . . . . . . . . .       (Blank, an SQLID, or NONE)
  Use DEFER YES . . . . . . . . . . . . NO    (Yes/No)
  VIEW Column List  . . . . . . . . . . YES   (Yes/No)
  Perform recovery analysis . . . . . . NO    (Yes/No)
  Enable authorization switching  . . .       (Yes/No)

Perform analysis in batch . . . . . . . YES   (Yes/No)

Show this panel prior to each use . . . YES   (Yes/No)

Change diagnostic options   . . . . . . NO    (Yes/No)
```

Figure 195. ALTER Analysis Options panel (ADBP7P)

6. On the ALTER - Build Analyze and Apply Job panel (ADBPALT) you can specify options for building the WSL or batch job that is used to implement the changes. After you have specified your options, press Enter to run the job.

```
ADBPALT ------------------ ALTER - Build Analyze and Apply Job ---------------
Option ===>

 Specify the following:
                                                            More:     +
 Worklist information:
   Worklist name . . . . . . .          (also used as middle qualifier in DSNs)
   Prefix for data sets  . . .

 Data set information:
   PDS final qualifiers  . . .
     Member name . . . . . . . ADBALTER
     Delete member name  . . . ADBDELET (Optional job to delete work data sets)

 Options:
   Generate online . . . . . . NO       (Yes/No)
   Generate one job  . . . . . YES      (Yes/No)
     Member name or prefix . . APPLY
   As work statement list  . . YES      (Yes/No)
   Unload method . . . . . . . U        (Unload, Parallel unload, HPU)
   Authorization Switch ID . .          (SQLID to connect, <SQLID>, blank,
                                          or NONE)
   SECADM Authorization ID . .          (SQLID to connect or blank)

     Optional processes:
   Run CHECK DATA  . . . . . NO        (Yes/No)
   Run COPY  . . . . . . . . N         (after: Reload/Alter/Both/None)
   Run REORG/REBUILD . . . . N         (Mandatory, All relevant, None)
   Run RUNSTATS  . . . . . . N         (after: Reload/Alter/Both/None)
   Run REBIND  . . . . . . . NO        (Yes/No)

   Utility control options:
     Use templates . . . . . .          (Yes/No)
     Use utility options . . .          (Yes/No)

 BP - Change batch job parameters
 TU - Specify TEMPLATE usage
 UO - Customize utility options
 CO - Change options common to change functions
```

Figure 196. ALTER - Build Analyze and Apply Job panel (ADBPALT)

Examples of altering a table with the AL line command
The examples in this topic show how to alter a table with the AL line command.

Adding a primary key to a table:
About this task

To add a primary key to a table:

Procedure
1. From the Tables, Views, and Aliases panel (ADB21T), issue the AL line
   command against the table that you want to add a primary key to. The Alter
   Table panel (ADB21TA) is displayed.
2. Type an S before **Add primary key** and press Enter. The Add Primary Key
   Constraint panel (AB21TAN) is displayed. If you need help selecting the
   columns for the primary key, use the COLUMNS primary command to display
   a list of the columns. Use the sequence line command to specify a number for
   the relative position of each column that you want in the primary key. Press
   PF3 to return to the previous panel.
3. Optional: Specify a name for the primary key constraint.
4. Press Enter to run the ALTER TABLE statement. The primary key is created.

Adding a partitioning key to table:
Procedure

1. From the Tables, Views, and Aliases panel (ADB21T), issue the AL line command against the table that you want to add a partitioning key to. The Alter Table panel (ADB21TA) is displayed.

2. Type an S before ADD PARTITIONING KEY and press Enter. The Alter Table panel (ADB21TAP) is displayed.

3. Select the columns to be part of the partitioning key and their order (A is ascending, D is descending). You can also use the 3D line command to assign a specific column sequence. If you need to start over and eliminate the changes you make, use the ORIGINAL primary command.

4. Enter the CONTINUE primary command to display the Alter Partitioned Table panel (ADB21TAV). If you want to remove a particular column from the set of selected columns for the key, use the R line command. If you need help entering limit key values, use the COLUMNS primary command to list the details of the columns that are selected to be part of the key on the previous panel ADB21TAP.

5. After you enter the limit key values for all partitions, enter the CONTINUE primary command to run the ALTER TABLE statement and create a partitioning key. Panel ADB21TA is displayed again.

Adding a partition to a table:
Procedure

1. From the Tables, Views, and Aliases panel (ADB21T), issue the AL line command against the partitioned table that you want to add a partition to. The Alter Table panel (ADB21TA) is displayed.

2. Type an S before ADD PARTITION and press Enter. The Alter Partitioned Table panel (ADB21TAV) is displayed.

3. Issue the ADD primary command to add a row with the next partition number generated.

4. Enter the partition limit key values, according to the Partitioning index/Data partitioned secondary index that is already created for the table.

5. After entering the limit key values for the new partition, use the CONTINUE primary command to display the ALTER - STOP command confirmation panel (ADB21TAS).

6. Enter the appropriate choice. For example, option 1 runs the stop database statements, alters the table, and runs the start database statements. The partition is then added to the table.

7. Press Enter to run the ALTER TABLE statement.

Altering a partition:
Procedure

1. From the Tables, Views, and Aliases panel (ADB21T), issue the AL line command against the partitioned table that you want to alter a partition for. The Alter Table panel (ADB21TA) is displayed. When conditions are met, ADD/ALTER PART TABLE is included in the list of options.

2. Type an S before ADD/ALTER PART TABLE and press Enter. The Alter Partitioned Table panel (ADB21TAV) is displayed.

3. Change the limit key values for any of the partitions.

4. Use the CONTINUE primary command to run the ALTER TABLE statement. The partitions are altered with their new values.

Rotating a partition:
Procedure
1. From the Tables, Views, and Aliases panel (ADB21T), issue the AL line command against the partitioned table that you want to rotate partitions for. The Alter Table panel (ADB21TA) is displayed. When conditions are met, ADD/ALTER PART TABLE is included in the list of options.
2. Type an S before ADD/ALTER PART TABLE and press Enter. The Alter Partitioned Table panel (ADB21TAV) is displayed.
3. Use the ROTATE primary command to rotate a partition. A pop-up panel (ADB21TAV) is displayed.
4. Select Option 1 - Execute the statement on the ADB2PSTM panel. The Alter Table - Utilities panel (ADB21TAU) is displayed. The ROTATE statement is held until all the other ALTER statements are executed. If the first logical partition of the table space is in REORG, run the REORG utility before running ROTATE.
5. Press Enter. Press Enter and the JCL screen appears. The ALTER TABLE statement shows a successful rotate partition, as shown in the following example:

```
ALTER TABLE "SMITHJR"."TBADAJ01" ROTATE PARTITION FIRST TO LAST  ENDING
AT ('10500') RESET;
```

Example: Dropping a column:
About this task

To drop a column:

Procedure
1. From the main menu, select option T. The Tables, Views, and Aliases panel is displayed.
2. Issue the AL command against the table to be changed. The Alter Table panel, as shown in the following figure, is displayed.

```
ADB21TA n ---------------------- DSNB Alter Table ---------------------
Command ===>

                                                          More:     +
 Table schema . . : DSN81010  >
 Table name . . . : T1                 >

   AUDIT  . . . . . . . NONE      (None, Changes, or All)
   DATA CAPTURE . . . . NONE      (None/Changes)
   VALIDPROC  . . . . . NULL      (NULL/Program name)
   RESTRICT ON DROP . . NO        (Yes/No)
   VOLATILE . . . . . . NO        (Yes/No)
   APPEND . . . . . . . NO        (Yes/No)

 ALTER TABLE with any of the above changes OR select one of the options below

   ADD column                             ADD MATERIALIZED QUERY
 s DROP COLUMN                            DROP MATERIALIZED QUERY
   ADD PRIMARY KEY                        REFRESH MATERIALIZED TABLE
   DROP PRIMARY KEY                       ADD PARTITIONING KEY
   ADD FOREIGN KEY                        ADD PARTITION
   DROP FOREIGN KEY                       ADD CLONE
   ADD CHECK constraint                   DROP CLONE
   DROP CHECK constraint                  ADD VERSIONING
   ADD UNIQUE constraint                  DROP VERSIONING
   DROP UNIQUE constraint                 ADD PERIOD
   ADD ORGANIZE BY HASH                   ADD ROW PERMISSION
   ALTER ORGANIZATION                     DROP ROW PERMISSION
   DROP ORGANIZATION                      ADD COLUMN MASK
   ACTIVATE ROW ACCESS CONTROL            DROP COLUMN MASK
   DEACTIVATE ROW ACCESS CONTROL
   ACTIVATE COLUMN ACCESS CONTROL
   DEACTIVATE COLUMN ACCESS CONTROL
```

3. Issue the Drop Column command. The Columns in Table panel, as shown in the following figure, is displayed.

```
ADB21TC n -- DSNB Columns in Table DSN81010.T1         ------- Row 1 to 2 of 2
Command ===>                                           Scroll ===> PAGE
Select by typing 'DROP'
Line commands:
 T - Tables  X - Indexes  A - Auth  GR - Grant  H - Homonyms  I - Interpret
 UR - Update runstats  LAB - Label  COM - Comment  DI - Distribution stats
 ? - Show all line commands

Select Column Name        Col No Col Type Length Scale  Null Def FP    Col Card
       *                      * *              *     * *   *   *              *
------ ------------------ ------ -------- ------ ------ ---- --- -- -----------
DROP   C1                      1 INTEGER       4      0 N    N   N          -1
       C2                      2 CHAR          1      0 Y    Y   N          -1
****************************** END OF DB2 DATA ******************************
```

Restriction: The Drop Column command is not selectable from the Alter Table panel, if any of the following conditions is true:

• The table in not contained in a universal table space (UTS)

• The table is a materialized query table (MQT)

• The table is referenced in a MQT definition

• The table contains an edit procedure or a validation-exit procedure

• The table is in an incomplete state

• The table is a system-period temporal table.

• The table contains extended indexes that are dependent on the table

• The table contains triggers that are dependent on the table

- The table contains row permissions that are dependent on the table
- The table contains column masks that are dependent on the table
- The table contains check constraints that are dependent on the table

4. Issue the Drop line command against the column that you want to drop.

   Restriction: The Drop command can be issued against only one column at a time.

Examples of redefining a table

The examples in this topic show how to redefine a table.

Example: Inserting a column:
About this task

To insert a column into a table:

Procedure

1. From the main menu, select option T. The Tables, Views, and Aliases panel is displayed.
2. Issue the ALT command against the table to be changed. The Alter Table panel is displayed. (For information about any of the fields in this panel, access the online help.)
3. Issue the I line command, as shown in Figure 197, to insert a new column at the specified position.

```
DB2 Admin ------------------- DB2X ALTER Table ------------------ Row 1 from 5
Command ===>                                              Scroll ===> PAGE

New owner ===> DSN8810  >                  Old owner : DSN8810
New name  ===> DEPT              >         Old name  : DEPT
Partitions ==> 1         Action    : NONE  New DB    : DSN8D81A
Rows/Page    : 48.188                      New TS    : DSN8S81D
Commands :  NEXT CONSTRAINTS TBLOPTS          HASH
Line commands :
 I - Insert  U - Update  D - Delete  R - Repeat   LAB - Label  COM - Comment
 M - Move    A - After   B - Before  X - Index    RES - Reset update
 UM - Update XML modifiers
                                                       Old Operation
Select Column Name       Col No Col Type Length  Scale Null D Col No Type
       *                     *  *              *      * *    *      * *
------ ------------------ ------ -------- ------ ------ ---- - ------ ---------

       DEPTNO               1 CHAR         3        0 N    N      1
I      DEPTNAME             2 VARCHAR     36        0 N    N      2
       MGRNO                3 CHAR         6        0 Y    Y      3
       ADMRDEPT             4 CHAR         3        0 N    N      4
       LOCATION             5 CHAR        16        0 Y    Y      5
***************************** END OF DB2 DATA *****************************
```

Figure 197. Alter Table panel (ADB27C) - Inserting a column

4. Fill in the **?** fields on the **Operation Type** INSERT line as shown in the following figure, to define the new column and press Enter. The Alter Table panel is displayed again.

```
ADB27C in ------------------ VA1A ALTER Table -------------- Row 1 to 6 of 6
Command ===>                                              Scroll ===> PAGE

New schema . . DSN8A10   >                 Old schema: DSN8A10
New name . . . DEPT              >         Old name  : DEPT
Partitions . : 0                           New DB  . . DSN8DA1A
Rows per page: 53                          New TS  . . DSN8SA1D

Commands:  CONTINUE  CONSTRAINTS TBLOPTS        HASH
Line commands:
 I - Insert  U - Update  D - Delete  R - Repeat  LAB - Label  COM - Comment
 M - Move    A - After   B - Before  X - Index   RES - Reset update
 UM - Update XML modifiers
                                                             Old Operation
Sel Column Name        Col No Col Type     Length  Scale N D Col No Type
    *                    * *                 *      * * *     * *
--- ----------------->  ------ -------- ----------- ------ - - ------ ---------

    DEPTNO               1 CHAR           3        0 N N     1
*   DEPTNAME             2 VARCHAR        36       0 N N     2
    BUILDING             3 CHAR           0        0 ? ?     0 INSERT
    MGRNO                4 CHAR           6        0 Y Y     3
    ADMRDEPT             5 CHAR           3        0 N N     4
    LOCATION             6 CHAR           16       0 Y Y     5
***************************** END OF DB2 DATA *****************************
```

Figure 198. Alter Table panel (ADB27C) - Specifying attributes for the inserted column

To specify additional attributes for a new column, you can use the U line
command to display the Update Column panel, as shown in the following
figure. Press Enter to return to the Alter Table panel.

```
ADB26CTU  ---------------------- DSNA ALTER Table ---------------------- 10:08
Command ===>

                                                        More:     +
 DB2 Admin ALTER                    Schema . : DSN81010 >
                                    Name . . : DEPT                >
Column name  . . NEWCOL          >  (column number 2)
Column type  . . CHAR            (CHAR,DECIMAL,INTEGER,SMALLINT,etc.)
Data length  . . 1
Inline length  .                 (0-32680 BLOB or CLOB, 0-16340 DBCLOB)
Precision  . . .                 (FLOAT and DECIMAL only)
Scale  . . . . .                 (DECIMAL and TIMESTAMP only)
Type schema  . .                 (User-defined type schema)
Type name  . . .                 (User-defined type name)
CCSID  . . . . . 1208            (1208 VARCHAR, 1200 VARGRAPHIC)
WITH TIME ZONE .                 (Yes/No - for TIMESTAMP only)

Allow Nulls  . . NO   (Yes-Nullable, No-NOT NULL)
FOR ? DATA . . .      (B - Bit, S - SBCS, M - Mixed, or blank)
WITH DEFAULT . . NO   (Yes, No, L (SECLABEL) or enter value below)
Default value  .

GENERATED  . . .     (A-ALWAYS, D-DFLT, I-ALWAYS AS IDENT, J-DFLT AS IDENT,
                      E-ALWAYS AS UPD TIMESTAMP, F-DFLT AS UPD TIMESTAMP)
FIELDPROC
Program name . .
Program parm . .                                        >
```

Figure 199. Update Column panel (ADB26CTU)

5. Use the ALTER - Build Analyze and Apply Job panel (ADBPALT) panel to
   choose options for building the WSL or batch job used to implement the
   change. After you have entered your options, press **Enter** to run the job.

Example: Updating a column:

About this task

To update a column:

Procedure

1. From the main menu, select option T. The Tables, Views, and Aliases panel is displayed.
2. Issue the ALT command against the table to be changed. The Alter Table panel, as shown in the following figure, is displayed.

```
ADB27C in ------------------ DSN9 ALTER Table -------------- Row 1 to 5 of 5
Command ===>                                       Scroll ===> CSR

New schema . . BDB  >                    Old schema: DSN81010
New name . . . BDBCATVT          >       Old name  : DEPT
Partitions . : 1                         New DB  . . DSN8D10A
Rows per page: 47                        New TS  . . DSN8S10D

Commands :  NEXT CONSTRAINTS TBLOPTS         HASH
Line commands :
  I - Insert  U - Update  D - Delete  R - Repeat    LAB - Label  COM - Comment
  M - Move    A - After   B - Before  X - Index     RES - Reset update
  UM - Update XML modifiers
                                                           Old Operation
Sel Column Name        Col No Col Type    Length  Scale N D Col No Type
    *                    * *                  *     * * *    * *
--- ----------------->  ------ --------  ----------- ------ - - ------ ---------

    DEPTNO              1 CHAR            3       0 N N     1
    DEPTNAME            2 VARCHAR         36      0 N N     2
    MGRNO               3 CHAR            16      0 Y Y     3
    ADMRDEPT            4 CHAR            3       0 N N     4
U   LOCATION            5 CHAR            16      0 Y Y     5
****************************** END OF DB2 DATA ******************************
```

Figure 200. Alter Table panel (ADB27C) - Updating a column

3. Either type over the fields of the column that you want to update, or issue the U line command against the column. The U line command allows you to change more attributes. When you use the U line command, the Update Column panel, as shown in the following figure, is displayed. The current attributes for that column are displayed and are available for updating. (For information about any of the fields in this panel, access the online help.)

```
ADB26CTU  --------------------- DSNA ALTER Table --------------------- 10:27
Command ===>

                                                           More:     +
 DB2 Admin ALTER                      Schema . : DSN81010 >
                                      Name . . : DEPT                >
 Column name  . . LOCATION           >  (column number 6)
 Column type  . . CHAR               (CHAR,DECIMAL,INTEGER,SMALLINT,etc.)
 Data length  . . 16
 Inline length .                     (0-32680 BLOB or CLOB, 0-16340 DBCLOB)
 Precision  . . .                    (FLOAT and DECIMAL only)
 Scale . . . . .                     (DECIMAL and TIMESTAMP only)
 Type schema  . .                    (User-defined type schema)
 Type name  . . .                    (User-defined type name)
 WITH TIME ZONE .                    (Yes/No - for TIMESTAMP only)

 Allow Nulls  . . YES  (Yes-Nullable, No-NOT NULL)
 FOR ? DATA . . .      (B - Bit, S - SBCS, M - Mixed, or blank)
 WITH DEFAULT . . YES  (Yes, No, L (SECLABEL) or enter value below)
 Default value  . NULL

 GENERATED  . . .     (A-ALWAYS, D-DFLT, I-ALWAYS AS IDENT, J-DFLT AS IDENT,
                       E-ALWAYS AS UPD TIMESTAMP, F-DFLT AS UPD TIMESTAMP)
 FIELDPROC
```

Figure 201. Update Column panel (ADB26CTU)

4. Make your changes to the column and press Enter to redisplay the Alter Table panel.

   About primary key columns
   > If you are altering a primary key column of a table, an additional primary command, ADDFK, is available on panel ADB27C to propagate the primary key column updates for the target table to all tables affected by the update. All tables become part of the ALTER JCL and are displayed on the Alter Tables panel (ADB27CA).

   About identity columns
   > If you are altering a table that contains an identity column and the table is being dropped and re-created, the column definition becomes GENERATED BY DEFAULT to preserve current data values. The first value that will be generated for the identity column (the START WITH clause) is also changed. The new START WITH value, which is the value that will be assigned next to the identity column, is the last unassigned value (MAXASSIGNEDVAL in SYSIBM.SYSSEQUENCES) plus the increment value (INCREMENT in SYSIBM.SYSSEQUENCES). If values were cached, any existing unassigned values in the cache that have not been used are lost. Loss of unassigned cached values causes a gap between the last assigned value of the identity column and the new starting value.

5. Use the ALTER - Build Analyze and Apply Job panel (ADBPALT) panel to choose options for building the WSL or batch job used to implement the change. After you have entered your options, press **Enter** to run the job.

Example: Adding a unique key to a table when unique constraints (primary key and unique keys) already exist:
About this task

To add a unique key:

Procedure

1. From the Tables, Views, and Aliases panel (ADB21T), issue the ALT line command against the table that you want to add a unique key to. The Alter Table panel (ADB27C) is displayed.

2. Enter the CONSTRAINTS primary command to display the ALTER - Unique Constraints panel.

   This panel lists the primary key and unique key constraints for the table.

3. Enter the Add primary command. The Create Primary or Unique Key panel (ADBP7CTP) is displayed.

4. Specify the options for the unique key.

   a. Type a name for the constraint in the **Constraint name** field.

   b. Specify whether the key is a primary or a unique key in the **Type** field.

   c. For the columns in the table, use the nn line command to specify the relative position of the column in the key.

Example: Changing a unique key:
About this task

To change a unique key:

Procedure

1. From the Tables, Views, and Aliases panel (ADB21T), issue the ALT line command against the table that you want to add a unique key to. The Alter Table panel (ADB27C) is displayed.

2. Enter the CONSTRAINTS primary command to display the ALTER - Unique Constraints panel.

   This panel lists the primary key and unique key constraints for the table.

3. Enter the Alter (A) line command for the constraint that you want to change. Depending on the type of constraint, either the Alter Primary Key or Alter Unique Key panel (ADBP7CTP) is displayed.

4. Specify the options that you want to change.

   a. If you are changing a primary key, you can type a new name in the **Constraint name** field.

   b. Use the nn line command to change the relative position of the column in the key.

Example: Renaming a table:
About this task

To rename a table:

Procedure

1. From the main menu, select option T to display the Tables, Views, and Aliases panel.

2. Issue the ALT line command against the table that you want to rename.

   Tip: Another way to rename a table without using the ALT line command to redefine the table is to use the REN line command from the Tables, Views, and Aliases panel.
   The Alter Table panel is displayed.

3. Type the new name of the table in the New name field and press Enter.

4. On the Alter Tables panel (ADB27CA) enter `NEXT` on the command line of the panel.

5. Use the ALTER - Build Analyze and Apply Job panel (ADBPALT) panel to choose options for building the WSL or batch job used to implement the change. After you have entered your options, press **Enter** to run the job.

Example: Adding a partition to a table:
Procedure

1. From the Tables, Views, and Aliases panel (ADB21T), issue the ALT line command against the partitioned table that you want to add a partition to. The Alter Table panel (ADB27C) is displayed.

2. Use the ALTPART command on the command line to add or alter a partition of a table-based partitioned table and press Enter. On the Alter Partitioned Table panel (ADB27CPV) you can see the Partitions field is updated to reflect the change.

Example: Adding a partition to a table in a partition by growth table space:
Procedure

1. From the Tables, Views, and Aliases panel (ADB21T), issue the ALT line command against the partitioned table that you want to add a partition to. The Alter Table panel (ADB27C) is displayed.

2. Use the ADDPART command on the command line and press Enter. The Partitions field is updated to reflect the change. The number defaults to 1. Specifying a zero (ADDPART 0) resets the number of partitions to the original value.

Recovering a table if the change fails

If a table modification fails and the original table is dropped, you can restore the table to its original state.

About this task

To recover a table:

Procedure

1. Drop the new table if it has been created.
2. Re-create the original table using the extracted DDL.
3. Load the table by using the unload data set. Remember to change the LOAD utility statement to RESUME YES if other tables exist in the table space.
4. Create a new image copy of the table space.
5. Run RUNSTATS on the table.

How the DB2 Admin Alter ALT function works

When you use the ALT line command to change an object, such as a table, you invoke the DB2 Admin Alter ALT function.

After you specify your changes in the online dialogs, you use the ALTER primary command from the Alter Objects panel (ADB27CA) to generate the batch jobs that perform the actual alter operation.

Before the required batch jobs are generated, the ALTER - Build Analyze and Apply Job panel (ADBPALT) is displayed. On the ALTER - Build Analyze and Apply Job panel, you specify various information, such as:

- The worklist name
- The PDS where you want the various jobs placed and the prefix to be used for generated data sets (the prefix is not used if templates are used)
- The prefix to be used for generated data sets (the prefix is not used if templates are used)
- The member name of a single job, if you want to combine the generated jobs
- Optional steps to run after the table is redefined, such as REORG

You can also use the BP command to update the unit type and space parameters that are used for allocating the work and unload data sets.

When you press Enter on the Alter Parameters panel, the DB2 Admin Alter ALT function generates the jobs and displays a PDS EDIT session for the specified job PDS that contains the jobs. The generated jobs are:
- ST1RE - Reverse engineering
- ST1REALL - Reverse engineering when restore is enabled (optional)
- ST2ULnnn - Unload data for table nnn
- ST3AC - Convert
- ST4AR - Alter related merge DDL (optional)
- ST5DC - Drop and re-create
- ST6RLnnn - Reload data
- ST7CD - Check Data (optional)
- ST8ICnnn - Image Copy
- ST9RS - Runstats (optional)
- ST10RB - Rebind (optional)
- ST11DL - Delete work data sets, except those for restore and unload (optional)

DB2 Admin does not generate the conversion job step if it can determine that data conversion is not needed.

The numeric values in these job names are adjusted to occur sequentially if you omit optional steps or DB2 Admin determines that the conversion step is not needed. For example, if related objects are not included, ST5DC becomes ST4DC.

Review the jobs and submit the jobs in the sequence shown in the list of generated jobs to perform the changes.

If you choose to have the statements that are necessary to make the changes put in a work statement list (WSL) and specify that the WSL is to be generated online instead of with a batch job, JCL to create the WSL is generated and run online. Messages are displayed to indicate the status as each step is run. When the online processing is complete, a work data set is displayed. This work data set contains the messages that would be seen in the job output if the WSL had been generated with a batch job.

You can use authorization switching when you redefine tables if authorization switching is enabled on the subsystem.

Changing the related objects for a table

You can alter table spaces, databases, indexes, views, foreign keys, and many other objects that are related to one or more tables.

Before you begin

Ensure the System Catalog panel is displayed.

Procedure

1. From the main menu, select option T. The Tables, Views, and Aliases panel is displayed.
2. Issue the ALT command against the table whose related objects you want to change. The ALTER Table panel is displayed.
3. Issue the CONTINUE command. The ALTER Objects panel (ADB27CA) is displayed.
4. Optional: If the table object that you want to change is not shown, access the ALTER Choose Related Objects panel (ADBP7OBJ).
   a. Issue the OPTIONS command. The ALTER Options Menu panel (ADBP7OP) is displayed.
   b. Select option 2 REL options. The ALTER Options Menu panel (ADBP7OP) is displayed.
   c. Indicate YES for object that you want to view and then issue the CONTINUE command. You then exit until you return to the ALTER Objects panel (ADB27CA).
5. Issue the REL line command against the table that you want to change. The Related Objects panel is displayed, which shows the related objects for the table.

```
ADBP7REL  -------------- VA1A ALT - Related Objects -------- Row 1 to 17 of 17
Command ===>                                              Scroll ===> PAGE

Line commands:  S - Show object  A - Add object

Related objects for table:      DSN8A10.DEPT

Sel Type    Object Name            Qualifier Info 1   Info 2   Note
    *       *                      *         *        *        *
--- ------- ----------------------> --------- -------> -------> -------------
    D------ DSN8DA1A---------------- SYSADM
     S        DSN8SA1D              SYSADM                     Segmented
      T         DEPT                DSN8A10   DSN8DA1A DSN8SA1D
       Y          DEPT              SYSADM    DSN8A10  DEPT
      CHR         RDD                         DSN8A10  DEPT     Child
      CHR         RED                         DSN8A10  EMP      Child
      CHR         DEPTNO                      DSN8A10  PROJ     Child
      PAR         RDD                         DSN8A10  DEPT     Parent
      PAR         RDE                         DSN8A10  EMP      Parent
       X          XDEPT1            DSN8A10   DSN8A10  DEPT     Primary
       X          XDEPT2            DSN8A10   DSN8A10  DEPT
       X          XDEPT3            DSN8A10   DSN8A10  DEPT
       V          VDEPMG1           DSN8A10   DSN8A10  DEPT
       V          VDEPT             DSN8A10   DSN8A10  DEPT
       V          VEMPDPT1          DSN8A10   DSN8A10  DEPT
       V          VHDEPT            DSN8A10   DSN8A10  DEPT
       V          VPHONE            DSN8A10   DSN8A10  DEPT
```

Figure 202. Related Objects panel

6. Issue the A (alter) line command against the object that you want to change. The object is added to the Alter Objects panel.

Changing indexes

To change an index, you issue either the AL line command or the ALT line command against the index.

You can change an index in one of two ways:

- Use the AL line command to make certain changes that are supported by the ALTER INDEX statement. The Alter Index panel shows the changes can be made with only the AL command. DB2 Admin issues an ALTER INDEX statement to make the changes.
- Use the ALT line command to alter an index when the changes are more complex and are intrusive. An intrusive alter is one in which the objects have to be dropped and re-created, such as inserting a column in the middle of a table, dropping a column, renaming a column, or changing the attributes of a column.

  When you specify your index redefinition parameters, you can choose to save your request to a work statement list.

Altering an index

Use the AL line command to alter an index.

About this task

To alter an index with the AL line command:

Procedure

1. Enter the `al` line command against the index you want to alter under the **Select** column on the Indexes panel (ADB21X).

```
ADB21X in -------------------- DB2X Indexes ---------------- Row 1 to 1 of 1
Command ===>                                            Scroll ===> CSR

Commands: DIS  STA  STO  ALL XSPACE
Line commands:
 T - Tables  D - Database  G - Storage group  P - Plans  C - Columns
 DIS - Display index space  STA - Start index space  STO - Stop index space
 ? - Show all line commands

                        Index                 Table           C C C C
Select Index Name       Schema   Table Name   Schema   U  Cols G D L M
       *                *        *            *        *     * * * * *
------ ----------------- -------- ------------------ -------- - ------ - - - -
al    IXFGR             RIVERAF  TBFGR             RIVERAF  U     1 N N Y N
***************************** END OF DB2 DATA ******************************
```

Figure 203. Indexes panel (ADB21X)

2. Alter any index attributes and press Enter. DB2 Admin runs the SQL ALTER INDEX statement.

```
ADB21XA n ----------------- DB2X Alter Index --------------- Row 1 to 1 of 1
Command ===>                                              Scroll ===> CSR

Commands: ADDCOL

ALTER INDEX  RIVERAF.IXFGR                    (Nonpartitioned          )

Buffer Pool  . . . . BP1     Close Rule . . . . . YES  Copy Allowed  . . NO
Piece Size . . . . . 2097152 Cluster  . . . . . . NO   Padded  . . . . .
Compress . . . . . . NO

Sel   Part        Pqty  Sqty  FreePg %Free Erase ST VCAT    Stogroup GBPCache
--- ------ ----------- ------- ------ ----- ----- -- -------- ------->  --------

        0          -1     -1      0    10 NO     I  DSNA     SYSDEFLT CHANGED
***************************** END OF DB2 DATA ******************************
```

Figure 204. Alter Index panel (ADB21XA)

Results

For a partitioning index, a detail line is displayed for each partition. You can alter
any partition by updating the available attribute, such as %Free. To apply the same
change to all partitions of the index, provide a value in the All Parts row.

To change certain parameters, you must stop and restart the associated object. In
these cases, DB2 Admin issues a STOP table space or STOP index (or partition)
command and checks that the object is in a fully-stopped state. If stopped, it issues
an ALTER INDEX statement, followed by a START command. If the object is not in
a fully-stopped state, the STOP Check - Action panel prompts you to perform one
of the following actions:

• Check again
• Issue the ALTER
• Cancel the operation

If an object is not stopped when the ALTER TABLESPACE statement is run (for
example, if others are holding locks on the object), a -626 SQLCODE is displayed.

Renaming an index

Use the ALT line command to rename an index.

Procedure

1. In the Select column of the Indexes panel (ADB21X), enter the ALT line
   command against the index that you are renaming.

```
ADB21X in -------------------- DB2X Indexes ---------------- Row 1 to 1 of 1
Command ===>                                                  Scroll ===> CSR

Commands: DIS  STA  STO  ALL XSPACE
Line commands:
 T - Tables  D - Database  G - Storage group  P - Plans  C - Columns
 DIS - Display index space  STA - Start index space  STO - Stop index space
 ? - Show all line commands

                          Index                Table           C C C C
Select Index Name         Schema   Table Name  Schema   U  Cols G D L M
       *                  *        *           *        *      * * * *
------ ----------------- -------- ----------------- -------- - ------ - - - -
alt    IXFGR             RIVERAF  TBFGR           RIVERAF  U     1 N N Y N
****************************** END OF DB2 DATA ******************************
```

Figure 205. Indexes panel (ADB21X)

2. In the **CREATE INDEX** field, type over the original index name with the new name. Then, enter the NEXT primary command.

   In figure Indexes panel (ADB21X), you see that the original index name was IXFGR. In figure Redefine Index panel (ADB21XAR), you see that the index name was changed to IXFGRnew.

```
ADB21XAR  --------------- DB2X Redefine Index ------------------ Row 1 from 3
Command ===>                                                  Scroll ===> CSR

Commands: NEXT  ORIGINAL
Line commands: nnn A or D - Sequence and order  R - Remove the column
 A - Ascending  D - Descending  RA - Random   U - Update expression/XML pattern

CREATE INDEX RIVERAF    . IXFGRnew          >
         ON RIVERAF.TBFGR
Unique . . . . . . . YES     Where Not Null . . .    Cluster . . . . . NO
Buffer Pool  . . . . BP2     Close Rule . . . . . YES Copy Allowed  . . NO
Piece Size . . . . . 2097152 Define . . . . . . . YES Defer . . . . . .
Partitioned  . . . .         Padded . . . . . . .     Compress  . . . . NO

Select Column Name       Col Type    Length  Scale N ColSeq Ord OldSeq Ord
       *                 *               *      * *     * *         * *
------ ----------------->  -------- ----------- ------ - ------ --- ------ ---
       TIMESTAMP_GEN_ALWA TIMESTMP      10      6 N     1 A        1 A
       A                  INTEGER        4      0 N
       B                  CHAR           3      0 Y
****************************** END OF DB2 DATA ******************************
```

Figure 206. Redefine Index panel (ADB21XAR)

3. Enter the NEXT primary command on the command line of the Redefine Index - Space panel (ADB21XAS).

```
ADB21XAS  --------------- DB2X Redefine Index - Space -------- Row 1 to 1 of 1
Command ===>                                                  Scroll ===> CSR

Commands: NEXT  ORIGINAL
Line commands: O - Original data  C - Clear data

CREATE INDEX RIVERAF.IXFGRnew
       ON RIVERAF.TBFGR
Sel   Part       Pqty   Sqty FreePg %Free Erase ST VCAT    Stogroup GBPCache
      *          *      *     *      * *    * *     *        *
--- ------ ----------- ------ ------ ----- ----- -- -------- ------->  --------

       0          -1    -1     0    10 NO     I DSNA    SYSDEFLT CHANGED
****************************** END OF DB2 DATA ******************************

   File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
 sssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
```

Figure 207. Redefine Index - Space panel (ADB21XAS)

4. Enter the NEXT primary command on the command line of the ALTER Objects
   panel (ADB27CA).

```
ADB27CA n ----------------------- DB2X Alter Objects --------------- Row 1 of 1
Command ===>  NEXT                                              Scroll ===> PAGE

Commands: NEXT - Generate jobs    ADD - Add objects

  ALTOPT - Change alter options
Line commands:
  A - Alter Object  D - Delete  S - Select Object  REL - Alter related
  FK - Add Foreign Key-affected tables  RI - Add RI-related tables  E - Edit DDL
  RS - Reset RI-FK flags  CX - Create index  CFK - Create foreign key

    Object  Object                                  RI RI  FK
Sel Qual    Name               Ty Info 1   Info 2   Rels Add Add Operation
    *       *                  * *         *        * *   *   *
--- ------->  ------------------>  -- -------- -------- ------ --- --- ---------
    DSN81010 DEPT               TB PJOBTS   PJOBTS     5 NO  NO  NONE
****************************** END OF DB2 DATA ******************************
```

Figure 208. Alter Objects panel (ADB27CA)

5. Use the ALTER Analysis Options panel (ADBP7P) to select whether you want
   to do an online analysis or a batch analysis. Batch analysis is the preferred
   method.
   - To do a batch analysis, enter YES in the **Perform analysis in batch** field and
     press Enter.
   - To do an online analysis, enter NO in the **Perform analysis in batch** field and
     press Enter. If the analysis process determines that SQL ALTER statements
     will accomplish the task, panel ADB27CTC is displayed, which allows you to
     select whether you want to run the SQL statements in foreground (online) or
     to generate a batch job. If you select ALTER statements, the SQL is run in the
     foreground. If you specify batch jobs, panel ADBPALT is displayed. Also, if
     the analysis process determines that a DB2 ALTER statement cannot be used,
     panel ADBPALT is displayed.

```
ADBP7P in ---------------- DB2X ALTER Analysis Options ---------------- 14:30
Option ===>

Please specify the following for DB2 Admin ALTER:

Analysis options:
  Run SQLID . . . . . . . . . . . . . .          (Blank, an SQLID, or <NONE>)
  Object Grantor  . . . . . . . . . . .          (Blank or an SQLID)
  Use DEFER YES . . . . . . . . . . . . YES      (Yes/No)
  Retain GENERATED ALWAYS:
    For ROWID . . . . . . . . . . . .            (Yes/No)
    For ROW CHANGE TIMESTAMP  . . . . .          (Yes/No)
  IDENTITY START value  . . . . . . . .          (Original, Computed)
  SEQUENCE RESTART value  . . . . . . .          (Original, Computed)
  VIEW Column List  . . . . . . . . . . YES      (Yes/No)
  Perform recovery analysis . . . . . . NO       (Yes/No)
  Enable authorization switching  . . . YES      (Yes/No)

Perform analysis in batch . . . . . . . YES      (Yes/No)

Show this panel prior to each use . . . YES      (Yes/No)
```

Figure 209. ALTER Analysis Options panel (ADBP7P)

6. On the ALTER - Build Analyze and Apply Job panel (ADBPALT) you can
   specify options for building the WSL or batch job that is used to implement the
   changes. After you have specified your options, press Enter to run the job.

```
 ADBPALT ------------------ ALTER - Build Analyze and Apply Job ---------------
 Option ===>

 Specify the following:
                                                            More:    +
 Worklist information:
   Worklist name . . . . . . .            (also used as middle qualifier in DSNs)
   Prefix for data sets  . . .

 Data set information:
   PDS final qualifiers  . . .
     Member name . . . . . . . ADBALTER
     Delete member name  . . . ADBDELET (Optional job to delete work data sets)

 Options:
   Generate online . . . . . .            (Yes/No)
   Generate one job  . . . . . YES        (Yes/No)
     Member name or prefix . . APPLY
   As work statement list  . . YES        (Yes/No)
   Content of apply job(s) . . ALL        (All, DDL)
   Unload method . . . . . . . U          (Unload, Parallel unload, HPU)
   Authorization Switch ID . .            (SQLID to sign on as, blank or NONE)
   SECADM Authorization ID . .            (An ID to sign on as, blank or NONE)
   Disable REORG optimization . YES       (Yes/No)

   Optional processes:
     Run CHECK DATA  . . . . . NO          (Yes/No)
     Run COPY  . . . . . . . . N           (after: Reload/Alter/Both/None)
     Run REORG/REBUILD . . . . N           (Mandatory, All relevant, None)
     Run RUNSTATS  . . . . . . N           (after: Reload/Alter/Both/None)
     Run REBIND  . . . . . . . NO          (Yes/No)

   Utility control options:
     Use templates . . . . . .             (Yes/No)
     Use utility options . . .             (Yes/No)

 BP - Change batch job parameters
 TU - Specify TEMPLATE usage
 UO - Customize utility options
 CO - Change options common to change functions
```

Figure 210. ALTER - Build Analyze and Apply Job panel (ADBPALT)

Redefining an index or a partitioning index

Use the ALT command to redefine an index or a partitioning index.

About this task

The following classifications in the catalog (SYSINDEXES.INDEXTYPE) apply to indexes that are defined for a table. To understand these classifications, you should understand the difference between the terms *partitioned* and *partitioning*. The term *partitioned* describes an index that is physically partitioned into multiple data sets. The term *partitioning* describes an index that contains a superset of the partitioning columns of the table. One or many combinations of partitioned indexes (partitioning or non-partitioning) can be defined for a table, in addition to the traditional non-partitioned, secondary indexes.

Type 2

An index on a non-partitioned table or on a partitioned table that uses index-controlled partitioning.

Type P

Physically partitioned. A type P index is a partitioning, partitioned index that

contains columns that are a superset of the partitioning columns of the table, and that match the name, order, and sequence. Multiple partitioning indexes can exist for a single table.

Type D

Physically partitioned. A type D index (referred to as a DPSI) is a partitioned index that contains columns that are not a superset of the partitioning columns of the table. Multiple DPSIs can exist for any given partitioned table.

Procedure

1. In the Sel column of the Indexes panel (ADB21X), enter the ALT line command against the index you are redefining.

```
ADB21X in -------------------- DSNB Indexes --------------- Row 1 to 25 of 25
Command ===>                                          Scroll ===> CSR

Commands: DIS  STA  STO  ALL XSPACE
Line commands:
 T - Tables  D - Database  G - Storage group  P - Plans  C - Columns
 DIS - Display index space  STA - Start index space  STO - Stop index space
 ? - Show all line commands

                          Index                 Table           C C C C
Select Index Name         Schema   Table Name   Schema   U  Cols G D L M
       *                  *        *            *        *       * * * *
------ -----------------  -------- ------------ -------- - ------ - - - -
       IXFGRB             RIVERAF  TBFGRB       RIVERAF  U     3 N N Y N
       IXFGR              RIVERAF  TBFGR        RIVERAF  U     1 N N Y N
       IXFGR_PBR          RIVERAF  TBFGR_PBR    RIVERAF  U     1 N N Y N
ALT    IXFGR2             RIVERAF  TBFGR2       RIVERAF  U     1 N N Y N
       IXFGR2_PBR         RIVERAF  TBFGR2_PBR   RIVERAF  U     1 N N Y N
       IXFGRA             RIVERAF  TBFGRA       RIVERAF  U     1 N N Y N
       IXFGRI             RIVERAF  TBFGRI       RIVERAF  U     1 N N Y N
       IXFGRID            RIVERAF  TBFGRID      RIVERAF  U     1 N N Y N
       IXFGRID2           RIVERAF  TBFGRID2     RIVERAF  P     1 N N Y N
       IXFGRIX1           RIVERAF  TBFGRIX1     RIVERAF  D     2 Y Y Y N
       IXFGRMAS           RIVERAF  TBFGRMAS     RIVERAF  P     1 N N N N
       IXFGRMAT           RIVERAF  TBFGRMAT     RIVERAF  P     1 N N N N
       IXFGRMQ1           RIVERAF  TBFGRMQ1     RIVERAF  D     1 N N Y N
       IXFGRMQ2S1         RIVERAF  TBFGRMQ2S1   RIVERAF  D     1 N N Y N
       IXFGRMQ2S2         RIVERAF  TBFGRMQ2S2   RIVERAF  D     1 N N Y N
       IXFGRMQ2U          RIVERAF  TBFGRMQ2U    RIVERAF  D     1 N N Y N
       IXFGRTB2           RIVERAF  TBFGRTB2     RIVERAF  P     1 Y Y N N
       IXFGRTB4           RIVERAF  TBFGRTB4     RIVERAF  P     1 Y Y N N
       IXFGRXM5           RIVERAF  TBFGRXM5     RIVERAF  D     1 N N Y N
       IXFGRXM6           RIVERAF  TBFGRXM6     RIVERAF  D     1 N N Y N
       IXFGRXMP           RIVERAF  TBFGRXMP     RIVERAF  D     1 N N Y N
       IXFGRV             RIVERAF  TBFGRV       RIVERAF  U     1 N N Y N
       IXFGRV_PBR         RIVERAF  TBFGRV_PBR   RIVERAF  U     1 N N Y N
       IXFGRC             RIVERAF  TBFGRC       RIVERAF  P     1 N N Y N
       IXFGRG             RIVERAF  TBFGRG       RIVERAF  P     1 N N Y N
****************************** END OF DB2 DATA ******************************
```

Figure 211. Indexes panel (ADB21X)

2. Alter any index attributes and press Enter. Enter NEXT on the command line of the Alter Index panel (ADB21XAR).

```
ADB21XAR  --------------- DSNB Redefine Index ---------------- Row 1 to 2 of 2
Command ===>                                              Scroll ===> CSR

Commands: NEXT  ORIGINAL
Line commands: nnn A|D - Sequence & order  R - Remove the column  I - Include
 A - Ascending  D - Descending  RA - Random   U - Update expression/XML pattern
 B - Business Time without overlaps

CREATE INDEX RIVERAF    . IXFGR2               >
        ON RIVERAF.TBFGR2
Unique . . . . . . . YES      Where Not Null . . .    Cluster . . . . . NO
Buffer Pool  . . . . BP1      Close Rule . . . . . YES Copy Allowed  . . NO
Piece Size . . . . . 2097152 Define . . . . . . . YES Defer . . . . . .
Partitioned  . . . .          Padded . . . . . . .     Compress  . . . . NO
Exclude Null Keys  . NO

Select Column Name       Col Type     Length  Scale N ColSeq Ord OldSeq Ord
       *                 *              *       * *    * *       * *
------ ----------------- -------- ----------- ------ - ------ --- ------ ---
       A                 INTEGER        4       0 N     1 A       1 A
       B                 CHAR           3       0 Y
****************************** END OF DB2 DATA ******************************
```

Figure 212. Redefine Index panel (ADB21XAR)

3. Enter NEXT on the command line of the Redefine Index - Space panel
   (ADB21XAS).

```
ADB21XAS  --------------- DB2X Redefine Index - Space -------- Row 1 to 1 of 1
Command ===>                                              Scroll ===> CSR

Commands: NEXT  ORIGINAL
Line commands: O - Original data  C - Clear data

CREATE INDEX RIVERAF.IXFGRREN
        ON RIVERAF.TBFGR
Sel   Part      Pqty  Sqty FreePg %Free Erase ST VCAT      Stogroup GBPCache
       *          *     *     *     * *    * *    * *       * *       *
--- ------ ----------- ------ ------ ----- ----- -- -------- ------->  --------

      0           -1    -1     0    10 NO      I DSNA     SYSDEFLT CHANGED
****************************** END OF DB2 DATA ******************************
```

Figure 213. Redefine Index - Space panel (ADB21XAS)

4. Enter NEXT on the command line of the Alter Objects panel (ADB27CA).

```
 ADB27CA n ---------------------- DSNB Alter Objects ---------- Row 1 to 1 of 1
Command ===> NEXT                                         Scroll ===> CSR

Commands: NEXT - Generate jobs  ADD - Add objects
   ALTOPT - Change alter options
Line commands:
 A - Alter object  D - Delete  S - Select object  REL - Alter related
 FK - Add FK-affected tables  RI - Add RI-related tables  E - Edit view DDL
 RS - Reset RI-FK flags  CX - Create index  CFK - Create foreign key

    Object  Object                                 RI RI  FK
Sel Qual    Name              Ty Info 1  Info 2    Rels Add Add Operation
    *       *                 *  *       *         * *   *   *
--- ------> ----------------> -- ------> -------> ------ --- --- -----------
    RIVERAF IXFGR2            IX RIVERAF TBFGR2         NA  NA  MODIFY
****************************** END OF DB2 DATA ******************************
```

Figure 214. Alter Objects panel (ADB27CA)

5. Use the ALTER Analysis Options panel (ADBP7P) to select whether you want to do an online analysis or a batch analysis. Batch analysis is the preferred method.
   - To do a batch analysis, enter YES in the **Perform analysis in batch** field and press Enter.
   - To do an online analysis, enter NO in the **Perform analysis in batch** field and press Enter. If the analysis process determines that SQL ALTER statements will accomplish the task, panel ADB27CTC is displayed, which allows you to select whether you want to run the SQL statements in foreground (online) or to generate a batch job. If you select ALTER statements, the SQL is run in the foreground. If you specify batch jobs, panel ADBPALT is displayed. Also, if the analysis process determines that a DB2 ALTER statement cannot be used, panel ADBPALT is displayed.

```
ADBP7P in ----------------- DB2X ALTER Analysis Options ----------------- 14:30
Option ===>

Please specify the following for DB2 Admin ALTER:

Analysis options:
  Run SQLID . . . . . . . . . . . . . .           (Blank, an SQLID, or <NONE>)
  Object Grantor  . . . . . . . . . . .           (Blank or an SQLID)
  Use DEFER YES . . . . . . . . . . . . YES       (Yes/No)
  Retain GENERATED ALWAYS:
    For ROWID . . . . . . . . . . . .               (Yes/No)
    For ROW CHANGE TIMESTAMP  . . . . .             (Yes/No)
  IDENTITY START value  . . . . . . . .           (Original, Computed)
  SEQUENCE RESTART value  . . . . . . .           (Original, Computed)
  VIEW Column List  . . . . . . . . . . YES       (Yes/No)
  Perform recovery analysis . . . . . . NO        (Yes/No)
  Enable authorization switching  . . . YES       (Yes/No)

Perform analysis in batch . . . . . . . YES       (Yes/No)

Show this panel prior to each use . . . YES       (Yes/No)
```

Figure 215. ALTER Analysis Options panel (ADBP7P)

6. On the ALTER - Build Analyze and Apply Job panel (ADBPALT) you can specify options for building the WSL or batch job that is used to implement the changes. After you have specified your options, press Enter to run the job.

```
ADBPALT ------------------ ALTER - Build Analyze and Apply Job ---------------
Option ===>

 Specify the following:
                                                             More:     +
 Worklist information:
   Worklist name . . . . . . .            (also used as middle qualifier in DSNs)
   Prefix for data sets  . . .

 Data set information:
   PDS final qualifiers  . . .
     Member name . . . . . . . ADBALTER
     Delete member name  . . . ADBDELET (Optional job to delete work data sets)

 Options:
   Generate online . . . . . .            (Yes/No)
   Generate one job  . . . . . YES        (Yes/No)
     Member name or prefix . . APPLY
   As work statement list  . . YES        (Yes/No)
   Content of apply job(s) . . ALL        (All, DDL)
   Unload method . . . . . . . U          (Unload, Parallel unload, HPU)
   Authorization Switch ID . .            (SQLID to sign on as, blank or NONE)
   SECADM Authorization ID . .            (An ID to sign on as, blank or NONE)
   Disable REORG optimization . YES       (Yes/No)

   Optional processes:
     Run CHECK DATA  . . . . . NO         (Yes/No)
     Run COPY  . . . . . . . . N          (after: Reload/Alter/Both/None)
     Run REORG/REBUILD . . . . N          (Mandatory, All relevant, None)
     Run RUNSTATS  . . . . . . N          (after: Reload/Alter/Both/None)
     Run REBIND  . . . . . . . NO         (Yes/No)

   Utility control options:
     Use templates . . . . . .            (Yes/No)
     Use utility options . . .            (Yes/No)

 BP - Change batch job parameters
 TU - Specify TEMPLATE usage
 UO - Customize utility options
 CO - Change options common to change functions
```

Figure 216. ALTER - Build Analyze and Apply Job panel (ADBPALT)

Example of redefining an index: Excluding null keys

To save index space and to improve INSERT and query performance, you can
redefine an index from one that contains null keys to one that does not.

About this task

The index that is the subject of this task was originally created with null keys.

Procedure

1. In the Select column of the Indexes panel (ADB21X), issue the ALT line
   command against the index that you are redefining. The Redefine Index panel
   is displayed.

```
ADB21XAR  -------------- DSNB Redefine Index ---------------- Row 1 to 2 of 2
Command ===>                                            Scroll ===> CSR

Commands: NEXT  ORIGINAL
Line commands: nnn A|D - Sequence & order  R - Remove the column  I - Include
 A - Ascending  D - Descending  RA - Random   U - Update expression/XML pattern
 B - Business Time without overlaps

CREATE INDEX RIVERAF    . IXFGR2             >
         ON RIVERAF.TBFGR2
Unique . . . . . . . YES     Where Not Null . . .     Cluster . . . . . NO
Buffer Pool  . . . . BP1     Close Rule . . . . . YES Copy Allowed  . . NO
Piece Size . . . . . 2097152 Define . . . . . . . YES Defer . . . . . .
Partitioned  . . . .         Padded . . . . . . .     Compress  . . . . NO
Exclude Null Keys  . NO

Select Column Name       Col Type    Length  Scale N ColSeq Ord OldSeq Ord
       *                 *               *     * *     * *       * *
------ ----------------- -------- ----------- ------ - ------ --- ------ ---
       A                 INTEGER        4       0 N     1 A       1 A
       B                 CHAR           3       0 Y
***************************** END OF DB2 DATA *****************************
```

Note: The index contains null keys because the Exclude Null Keys attribute is set to NO.

Figure 217. Redefine index panel (ADB21XAR)

2. On the Redefine Index panel, type YES in the **Exclude Null Keys** field and press Enter. Issue the NEXT primary command.

```
ADB21XAR  -------------- DSNB Redefine Index ---------------- Row 1 to 2 of 2
Command ===>                                            Scroll ===> CSR

Commands: NEXT  ORIGINAL
Line commands: nnn A|D - Sequence & order  R - Remove the column  I - Include
 A - Ascending  D - Descending  RA - Random   U - Update expression/XML pattern
 B - Business Time without overlaps

CREATE INDEX RIVERAF    . IXFGR2             >
         ON RIVERAF.TBFGR2
Unique . . . . . . . YES     Where Not Null . . .     Cluster . . . . . NO
Buffer Pool  . . . . BP1     Close Rule . . . . . YES Copy Allowed  . . NO
Piece Size . . . . . 2097152 Define . . . . . . . YES Defer . . . . . .
Partitioned  . . . .         Padded . . . . . . .     Compress  . . . . NO
Exclude Null Keys  . YES

Select Column Name       Col Type    Length  Scale N ColSeq Ord OldSeq Ord
       *                 *               *     * *     * *       * *
------ ----------------- -------- ----------- ------ - ------ --- ------ ---
       A                 INTEGER        4       0 N     1 A       1 A
       B                 CHAR           3       0 Y
***************************** END OF DB2 DATA *****************************
```

Figure 218. Redefine index panel (ADB21XAR) - Redefining Exclude Null Keys attribute

3. On the Redefine Index - Space panel, issue the NEXT primary command. The Alter Objects panel is displayed.
4. On the Alter Objects panel, issue the NEXT primary command.

```
ADB27CA n --------------------- DSNB Alter Objects ---------- Row 1 to 1 of 1
Command ===> NEXT                                              Scroll ===> CSR

Commands: NEXT - Generate jobs  ADD - Add objects


   ALTOPT - Change alter options
Line commands:
 A - Alter object  D - Delete  S - Select object  REL - Alter related
 FK - Add FK-affected tables  RI - Add RI-related tables  E - Edit view DDL
 RS - Reset RI-FK flags  CX - Create index  CFK - Create foreign key

    Object  Object                                     RI RI  FK
Sel Qual    Name                Ty Info 1   Info 2    Rels Add Add Operation
     *       *                   *  *        *          *  *   *   *
--- ------->  ----------------->  -- ------->  ------->  ------ --- --- -----------
    RIVERAF  IXFGR2              IX RIVERAF  TBFGR2         NA  NA  MODIFY
****************************** END OF DB2 DATA ******************************
```

Figure 219. Alter Objects panel (ADB27CA)

The ALTER Analysis Options panel (ADBP7P) is displayed.

5. On the ALTER Analysis Options panel, type YES in the **Perform analysis in batch** field and press Enter. The ALTER - Build Analyze and Apply Job panel (ADBPALT) is displayed.

6. On the ALTER - Build Analyze and Apply Job panel, specify the options for building the WSL or batch job that is used to implement the changes and press Enter.

   The following panel provides an example of options that you might specify.

```
ADBPALT ------------------ ALTER - Build Apply Job -------------------------
Option ===>

 Specify the following:

 Worklist information:
   Worklist name . . . . . . . TEST2    (also used as middle qualifier in DSNs)
   Prefix for data sets  . . . RIVERAF

 Data set information:
   PDS final qualifiers  . . . TEST2.JCL
     Member name . . . . . . . ADBALTER
     Delete member name  . . . ADBDELET (Optional job to delete work data sets)

 Options:
   Generate online . . . . . . NO       (Yes/No)
   Generate one job  . . . . . YES      (Yes/No)
     Member name or prefix . . APPLY
   As work statement list  . . NO       (Yes/No)
   Content of apply job(s) . . ALL      (All, DDL)
   Unload method . . . . . . . U        (Unload, Parallel unload, HPU)
   Authorization Switch ID .. <NONE>    (SQLID to sign on as, blank or <NONE>)
   SECADM Authorization ID ..           (SQLID to sign on as, blank or <NONE>)
   Disable REORG optimization . YES      (Yes/No)

   Optional processes:
     Run CHECK DATA  . . . . . NO       (Yes/No)
     Run COPY  . . . . . . . . N        (after: Reload/Alter/Both/None)
     Run REORG/REBUILD . . . . N        (Mandatory, All relevant, None)
     Run RUNSTATS  . . . . . . N        (after: Reload/Alter/Both/None)
     Run REBIND  . . . . . . . NO       (Yes/No)

   Utility control options:
     Use templates . . . . . . NO       (Yes/No)
     Use utility options . . . NO       (Yes/No)

 BP - Change batch job parameters
 TU - Specify TEMPLATE usage
 UO - Customize utility options
```

Figure 220. ALTER - Build Analyze and Apply Job panel (ADBPALT)

Changing views

To make changes to a view, DB2 Admin generates a set of jobs that drop the view and then re-create it.

About this task

Restriction: You cannot use an SQL ALTER statement to change a view.

To change a view:

Procedure

1. Use the ALT line command on the Tables, Views, and Aliases panel (ADB21T). Panel ADB27CAA might appear briefly while the definition of the view is being retrieved. An SQL CREATE VIEW statement for the view is displayed in an ISPF Edit Session.
2. Edit the CREATE VIEW statement to make the changes that you want and press PF3. The Alter Tables panel (ADB27CA) is displayed. It shows an action of DROP.

If you did not change the CREATE VIEW statement or did not save the changes, the view either is not displayed on the Alter Tables panel or is displayed with an action of NONE.

3. Enter the ALTER command to display the Alter Parameters panel.

4. Use the ALTER - Build Analyze and Apply Job panel (ADBPALT) panel to choose options for building the WSL or batch job used to implement the change. After you have entered your options, press **Enter** to run the job.

Changing foreign keys

To make changes to foreign key attributes, you issue the ALT line command against the foreign key.

About this task

To change a foreign key:

Procedure

1. From the main menu, select option T to display the Tables, Views, and Aliases panel.

2. Issue the FK line command against a table to display the Foreign Keys panel, which shows the foreign keys for the table.

```
ADB21T in --------------- DB2X Tables, Views, and Aliases --------- Row 1 of 1
Command ===>                                              Scroll ===> PAGE

Commands: GRANT  MIG  ALL
Line commands:
 C - Columns  A - Auth  L - List  X - Indexes  S - Table space  D - Database
 V - Views  T - Tables  P - Plans  Y - Synonyms  SEL - Select prototyping
 ? - Show all line commands

Sel   Name             Schema    T DB Name  TS Name   Cols       Rows Chks C
      *                *         * *        *         *             *   * *
----- ----------------- -------- - -------- -------- ------ ----------- ---- -
FK    DEPT             DSN91010 T DSN9D10A DSN9S10D     5          14    0
***************************** END OF DB2 DATA ******************************
```

Figure 221. Tables, Views, and Aliases (ADB21T)

3. Issue the ALT line command against the foreign key that you want to change.

```
ADB21TFK  -- DB2X Foreign Keys of Table DSN91010.DE > ------------ Row 1 of 2
Command ===>                                              Scroll ===> PAGE

Line commands:
 FC - From Column   TC - To Column   T - To Table   ALT - Alter FK
    From:                To:
Sel Column Name       Rel Name Schema   Name             Column Name
    *                 *        *        *                *

--- ----------------- -------- -------- ----------------- -------------------
alt ADMRDEPT          RDD      DSN91010 DEPT             DEPTNO
    MGRNO             RDE      DSN91010 EMP              EMPNO
***************************** END OF DB2 DATA ******************************
```

Figure 222. Foreign Keys panel (ADB21TFK) - Changing a foreign key

4. Make changes to the foreign key attributes.

```
 ADB21TAF  -------------- DB2X Alter Foreign Key Constraint -------------- 08:20
 Command ===>

 Commands: COLUMNS

                                                           More:     +
  ALTER TABLE

  Table schema . . .  DSN91010   >
  Table name . . . .  DEPT              >

  FOREIGN KEY
  Constraint name  . . RDD01                  > (? to look up existing constraints

  Columns
         (   ADMRDEPT,MGRNO


                                                                    > )


  REFERENCES     Table schema . . . DSN91010  >
                 Table name . . . . DEPT             > (? to look up

  ON DELETE . . . .           (RESTRICT, CASCADE, SET NULL, or NO ACTION)
```

Figure 223. Alter Table (ADB21TAF)

5. Press Enter to return to the Alter Tables panel (ADB27CA).
6. Use the ALTER - Build Analyze and Apply Job panel (ADBPALT) panel to choose options for building the WSL or batch job used to implement the change. After you have entered your options, press **Enter** to run the job.

Using authorization switching

Authorization switching is a facility within DB2 Admin that is used to execute DDL and DCL under the authority of another user. The facility does not cover other statement types, including DB2 Utility commands and DSN subcommands such as FREE PACKAGE and BIND PLAN.

This other user is termed the *auth-switch ID*, and the ID that submits the job is termed the *submitter*.

Alter Tablespace ALT, Alter Table ALT, WSLs, Change Management, Change Management batch, and DB2 Object Comparison Tool make use of authorization switching. These commands enable you to redefine a table space or a table. Because this action also requires the object to be dropped, objects that are dependent on the target object are also dropped. Authorizations to those objects and dependent objects are lost.

DB2 Admin generates the DDL and DCL necessary to rebuild the altered objects and to restore the dependent objects and authorities. However, the job submitter might have authority to re-create the target objects, but not the authority to re-create dependent objects or to grant authorities to the dependent objects. In this case, you can enable a job submitter to use an ID that has the necessary authority to execute the DDL to rebuild the objects.

The statements that you can run with the auth-switch ID depend on your authority as defined in the RACF profile that protects the resource. If you have READ authority, the authorization switching function follows these rules and protections:
- Only certain DDL statements can be run using the auth-switch ID. ALTER (TABLE, MASK, PERMISSION, FUNCTION, and TRIGGER), COMMENT,

LABEL, CREATE, SET, GRANT (except system privilege) are auth-switch eligible statements. DROP statements, for example, are always run using the submitter's authority.
- Any DDL that has been manually added to the file or that has been edited can be run only under the submitter's authority.
- COMMIT statements can be added where appropriate.
- The DDL must be run within 8 days of being created.
- If ineligible statements are encountered, DB2 Admin will switch out of the requesting auth-switch ID and back into the auth-switch ID when an eligible statement is encountered.

If the job submitter has ALTER authority to the RACF profile that protects the resource, all DDL and DCL statements are run using the auth-switch ID. The rules and protection mechanisms for READ authority do not apply for ALTER authority.

When authorization switching is enabled, the batch job panels for Alter Tablespace AL and Alter Table ALT have an additional input field called **Authorization Switch ID** . Use this field to enter the auth-switch ID to be used to run the eligible statements in the file that contains the DDL and DCL statements.

The DDL that is generated by the batch jobs for these two functions is prepared for authorization switching; that is, it contains functional comments that other DB2 Admin components use with authorization switching.

The value <SQLID> is similar to the value SQLID except that you do not need to specify a specific auth-switch ID. If the value <SQLID> is specified in the Authorization Switch ID field, SQLID auth-switching DDL is generated. The SQLID auth-switching DDL automatically implements authorization switching. When the WSL runs, DB2 Admin can use dynamic auth-switching to drop and re-create objects by using the existing owner of the objects. During the running of a procedure, the WSL detects the need for authorization switching and provides the required auth-switch ID.

If the special value <NONE> is specified in the **Authorization Switch ID** field, the DDL is not prepared to be used with authorization switching, but an authorized ID can run the DDL. For example, the authorized ID can run the DDL using ADBTEP2.

If an authorization switch ID is not specified, and you specify Y in **As work statement list**, the work statement list does not produce DDL that is capable of authorization switching.

Tip: Carefully preserve the original DDL file until the objects and dependencies are restored. After the object is dropped, the ADB2GEN process cannot be used to regenerate the original environment. Running the ADB2GEN step again without proper care can overwrite the original DDL file, making reconstruction difficult.

The batch program, ADBTEPA, runs the DDL, either under the authority of the submitter or under the auth-switch ID authority. Two input parameters are required for authorization switching. These parameters are specified one-per line on the ADBOPT DD card in the ADBTEPA step.

Example

```
//CREATE    EXEC PGM=ADBTEPA,DYNAMNBR=100,
//  PARM='/SSID(DSN8),WORKLIST(TESTYA)'
//STEPLIB  DD DISP=SHR,DSN=ADBB10.SADBLINK
//         DD DISP=SHR,DSN=DSN810.SDSNEXIT
//         DD DISP=SHR,DSN=DSN810.SDSNLOAD
//SYSTSPRT DD SYSOUT=*
//ADBPRINT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//ADBOPT   DD *
 PLAN=ADBTEPA
 AUTH_SWITCH_USERID=SYSADMZ1
/*
```

The PLAN parameter is required by ADBTEPA, even when an auth-switch ID is not provided on the batch job panels. The AUTH_SWITCH_USERID parameter is generated, either as functional input when an ID is provided on the panel, or it is a comment without a value. If DB2 Admin Authorization Switching is determined to be necessary after the JCL is built, you can make the parameter active (remove the comment) and specify a suitable auth-switch ID.

To use DB2 Admin Authorization Switching, the job submitter must have access to the following two separate entities:

- The plan that is passed to program ADBTEPA using the ADBOPT parameter PLAN
- A RACF profile that protects a special resource

If the submitter has READ authority to the RACF profile, only certain DDL statements are executed using the authorization switch ID authority. DROP statements, for example, are always executed using the submitter's authority. If the job submitter has ALTER authority to the RACF profile that protects the resource, all DDL and DCL statements are run using the auth-switch ID. The rules and protection mechanisms that apply to READ authority do not apply to ALTER authority.

Only certain DDL statements are executed using the authorization switch ID authority. DROP statements, for example, are always executed using the submitter's authority. Any DDL that has been manually added to the file or that has been edited can be executed only under the submitter's authority. COMMIT statements can be added where appropriate.

Because the DDL contains SET CURRENT SQLID statements, the ID that runs the statement must have the intended SQLID as one of its secondary authorization IDs, or have SYSADM authority. If you want to suppress the generation of SET CURRENT SQLID statements, specify the value for Run SQLID as <NONE>. You can specify a RUN SQLID value in DB2 Admin functions such as GEN, ALT, Migrate, and Change Management.

Tip: Provide the authorization switch ID with SYSADM authority to successfully execute all statements within the DDL file, including the SET CURRENT SQLID statements.

Tip: When you specify <SQLID> as the auth-switch ID, the RUN SQLID field must be blank.

Tip: If you use the authorization switch with DB2 sample sign on exit, you must complete one of the following steps:

- Run the job by using an authorization switch ID that has SYSADM authority or that is connected to a group that has SYSADM authority.
- Define the owner of the objects as a RACF group and then run the job by using an authorization switch ID that is connected to the group ID.

For either of the preceding steps, define the authorization switch ID as a RACF user ID without a known password and with a password that never expires.

Using SQLID authorization switching

SQLID *Authorization switching* enables you to use authorization switching for tasks that require two or more authorization IDs. SQLID *Authorization switching* can be used with WSLs, Change Management, Change Management batch, DB2 Object Comparison Tool, and ALT.

To use SQLID auth-switching, you must specify an <SQLID> instead of a regular authorization ID. When you specify an <SQLID> as the auth-switch ID, the RUN SQLID field must be blank.

The SQLID authorization switching function follows these rules and protections:
- The SQLID must be verified for SET CURRENT SQLID to be executed.
- To verify the SQLID, execute the following statement using the submitter's ID:

  SELECT VERIFY_GROUP_FOR_USER(USER,:newsqlid) FROM SYSIBM.SYSDUMMY1

  The new sqlid is the ID used in SET CURRENT SQLID. When the SQLID you specify is verified, the statement returns 1, and SET CURRENT SQLID is executed
- If the ID is not verified, all statements will be executed using the submitter's ID, with these exceptions: CREATE and SET CURRENT SQLID statements.
- The CREATE statement will use the ID that was used in the last SET CURRENT SQLID statement. The ID can be a verified ID or a non-verified ID. If no SET CURRENT SQLID is provided, the submitter's ID is used.
- If a statement fails to execute, an authorization error such as -551 is issued, and the operation is performed using the object owner's authorization ID. There are exceptions where the object owner's authorization ID is not used as the retry ID. One exception is when the retry ID is used for GRANT and REVOKE on a view that uses the last ID specified in a PATHSCHEMAS. The other exception is when a retry ID is used for a CREATE statement.
- A CREATE statement retry is performed for Alias, Table, and View objects only when the object is qualified. The retry will use the submitter's ID as the authorization ID. Retry ID for a view uses the last ID specified in the CURRENT PATH or qualifier.
- If no SET CURRENT SQLID is provided in a WSL, ADBTEP2 will use the submitter's ID for alias, view, and table objects only when the object name is qualified; the retry for this case will use a qualifier as an authorization ID.

Using WSL authorization switching

WSL *Authorization switching* enables you to run a regular WSL using an auth-switch ID. You can only execute WSL auth-switching from the WSL pane.

Procedure

1. On the Change Options Common to Change Functions panel (ADB2PCO), set Enable WSL authorization switching to YES.

2. On the Work Statement List Library panel (ADB2W1), enter the R (RUN) line command to run a WSL. The Authorization Parameters panel (ADBPWLA) will appear.

3. On the Authorization Parameters panel, enter an *auth-switch ID* that you want to use as the primary authorization ID. When you specify the *auth-switch ID*, JCL that is eligible for WSL authorization switching is generated.

   Note: You must have ALTER authority to the RACF profile to use the WSL authorization switching function. You cannot use SQLID auth-switching along with WSL auth-switching.

Implicit LOB and XML table support

The DB2 Admin ALT and MIG functions and DB2 Object Comparison Tool support changes to implicit LOB and XML table spaces.

The DB2 Admin ALT, and MIG functions and DB2 Object Comparison Tool generate multiple image copies when there are implicit LOB or XML table spaces defined for the tables. Generating multiple image copies requires that either a SYSCOPY TEMPLATE is defined and used for the operations, or that the default is used. If no template is provided, this default is used:

```
DSN(&US..&SSID..&DB..&SN..&UQ)
```

The DB2 Admin ALT and MIG functions and DB2 Object Comparison Tool generate multiple image copies when there are implicit LOB or XML table spaces defined for the tables. Generating multiple image copies requires that either a SYSCOPY TEMPLATE is defined and used for the operations, or that the default is used. If no template is provided, this default is used:

```
DSN(&US..&SSID..&DB..&SN..&UQ)
```

clones, this default is used:

```
DSN(&US..&SSID..&DB..&SN..CLONE.&UQ)
```

Chapter 14. Migrating DB2 objects, data, views, and catalog statistics

DB2 Admin enables you to migrate (or copy) DB2 object definitions, the data for the objects, views, and the catalog statistics for the objects from one DB2 subsystem to other DB2 subsystems.

You can migrate any combination of this set of information (object definitions, data, views, and statistics) for DB2 databases, table spaces, and tables, as well as their dependent objects.

When you migrate information, DB2 Admin attempts to preserve as many of the dependent definitions as possible, such as indexes, views, table checks, synonyms, aliases, and authorizations to these objects.

Typically, the migrate function is used to perform the following tasks:

- Create a separate DB2 test system
- Move a test system into a production system
- Move statistics from a production system to a development (or test system) to test new and modified programs with the statistics from the production system.
- Consolidate two separate database systems into one

The following five steps summarize the process of using the DB2 Admin Migrate function:

1. Specify the information that you want to migrate. You can migrate object definitions, the data in the objects, the views, the catalog statistics for the objects, or any combination of these information sets.

   When you specify to migrate catalog statistics, INSERT, UPDATE, and DELETE statements that will modify the catalog statistics are generated. The statements are generated with the qualifier of the target catalog that you specify, and the statistic fields that are generated are those that are associated with the objects that are being migrated. (The complete list of statistics fields are those fields that are set by RUNSTATS that can be modified and the five statistics columns for table functions in SYSROUTINES, which are not set by RUNSTATS.)

2. Generate batch jobs.

   You can choose to have the migrate batch jobs generated online or in batch.

3. Run the batch jobs.

4. Optional: Transfer the jobs and data to the target system.

   You perform this step only when the node names for the source and target systems are different.

5. Run batch define and reload jobs and other optional jobs.

Each of these steps is described in detail later in this chapter.

In the event that one or more SQL statements fails when you are running a migrate job, you can use the Batch Restart program (ADBTEP2) to restart or resume the job at an intermediate point. In addition, you can combine the generated migrate batch jobs into fewer jobs.

Restrictions: The following restrictions apply to migrating DB2 object definitions, data, views, and catalog statistics:

- When a schema exists that is associated with a database, you must migrate the database structure and the schema separately.
- Databases without table spaces are not migrated. DB2 Admin issues a warning message that no rows are returned.
- For table spaces (or tables within these table spaces) that are created with the DEFINE NO option or for table spaces that are empty, you can migrate only the schema definitions (DDL). JCL or statements to unload the data are not created during migration.
- When migrating at the table level and migrating a table that has a LOB column, and the migrate option *DROP on target before create* is set to Yes, the base table will be dropped and DB2 will also drop any LOB auxiliary tables. Neither DB2 nor migrate will drop the LOB table space if they were explicitly defined and associated with each LOB auxiliary table. The user must drop the LOB table spaces.
- If the base table containing LOB column(s) is dropped and recreated, the explicit auxiliary table is recreated according to its source definition. Changes to the auxiliary table are not reported. Updates to the auxiliary table are ignored if the base table is not recreated.
- If you migrate the catalog statistics for the objects, the statistics for materialized query tables are not included.

Topics:
- "Step 1. Specify the objects to migrate or clone"
- "Step 2. Generate batch jobs" on page 340
- "Step 3. Run the batch jobs" on page 343
- "Step 4. Optional: Transfer the jobs/work statement list and data to the target system" on page 344
- "Step 5. Run the batch define, reload, and optional jobs" on page 345
- "Work data sets used by the DB2 Admin Migrate function" on page 345

Step 1. Specify the objects to migrate or clone

To begin migrating or cloning DB2 object definitions, the data for the objects, and the catalog statistics, first specify the objects. You can have either the DB2 Admin Tool migrate the objects or the DB2 Cloning Tool clone the objects.

Example

The starting point for migrating objects can be databases, table spaces, or tables. Issue the MIG primary command from the Database panel (ADB21D), Table Spaces panel (ADB28S) or Table panel (ADB21T).

You can specify object types that you want to include or exclude from the migration by using the GEN option.

```
ADB28M in ------------------- DB2X Migrate Parameters ------------------- 09:58
 Option ===> GEN

 Please specify the following for DB2 Admin Migrate:      DB2 System: DSNA
                                                          DB2 SQL ID: ISTJE
                                                                      More:    +
 Worklist name . . . . . . .              (also used as middle qualifier in DSNs)

 Data set information:
   PDS for jobs  . . . . . .   MYMIGR.JCL
   Prefix for datasets . . .   ISTJE

 Target system parameters:
   DB2 subsystem id (SSID) .   DB2X        DB2 release  . . . . : 1101
   Target system node name .              Submit job at local. : NO  (Yes/No)
   DB2 sample pgm load lib .   DBS.DSN110.RUNLIB.LOAD
   Target JCL job data sets for Admin and DB2
     Use customization settings for Admin libs . . . . . NO  (Yes/No)
     Use customization settings for the following libs . NO  (Yes/No)
       DB2 Admin APF library . .
       DB2 exit library  . . . .
       DB2 load library  . . .   SYS1.DSNDB2X.SDSNLOAD
   Catalog statistics options:
     Catalog qualifier . . . . . HI          > (default SYSIBM)
     Statistics tables . . . . . SELECT        (All or Select. Default is All)

 Migrate options:
   Generate MIG jobs in batch  . . .   NO      (Yes/No)
   Generate work stmt list . . . . .   NO      (Yes/No)
   Combine job steps . . . . . . . .   YES     (Yes/No, Yes if HPU Unload)
     Member prefix for combined jobs   ADBMG   (default ADBMG )
   Scope of migrate:
     DDL . . . . . . . . . . . . . .   N       (Yes/No)
     Data  . . . . . . . . . . . . .   N       (Yes/No)
     Catalog statistics  . . . . . .   N       (Yes/No)
   DROP on target before CREATE  . .   NO      (Yes/No,No if scope DDL is NO)
   Unload method . . . . . . . . . .   U       (U - Unload, H - HPU, C - Cross)
   Parallel utilities  . . . . . . .   NO      (Yes/No)

 Optional steps after reload:
   Run CHECK DATA  . . . . . . . . . : NO      (Yes/No)
   Run RUNSTATS  . . . . . . . . . . : NO      (Yes/No)
   Run IMAGE COPY  . . . . . . . . . : NO      (Yes/No)
   Run REBIND  . . . . . . . . . . . : NO      (Yes/No)

 Utility control options:
   Generate template statements  . . :         (Yes/No)
   Use customized utility options  . :         (Yes/No)

 BP  - Change batch job parameters
 TU  - Specify template usage
 UO  - Customize utility options
 GEN - GEN options
```

Figure 224. Migrate Parameters panel (ADB28M)

Tip: You can control the number of generated statistics. In the field **Statistics tables**, specify SELECT. Remove the **/** (slash) that is next to catalog tables that you do not want GEN to generate statistics for. For example, removing the **/** (slash) that is next to the SYSCOLDISTSTATS table turns off the generation of updates to the table.

Tables with the suffix of DISTSTATS are used to store partition-level statistics. The tables are not used by the optimizer, but are used by RUNSTATS. Therefore, tables with the suffix of DISTSTATS can be turned off if you do not plan to run RUNSTATS on the target objects.

```
ADBP8MG n -------------- DSNB Generate SQL from DB2 catalog ------------- 23:44
Command ===>

Show this panel prior to each use . . N (Y,N)

SQL statement types to be generated from the DB2 catalog:
  CREATE VIEW  . . . . . . . . . . Y (Y,N,D)
  CREATE INDEX . . . . . . . . . . Y (Y,N)
  CREATE SYNONYM . . . . . . . . . Y (Y,N)
  CREATE ALIAS . . . . . . . . . . Y (Y,N)
  CREATE TRIGGER . . . . . . . . . Y (Y,N,D)
  CREATE MASK  . . . . . . . . . . Y (Y,N)
  CREATE PERMISSION  . . . . . . . Y (Y,N)
  CREATE STORAGE GROUP . . . . . . Y (Y,N)

  GRANT access ON DATABASE . . . . Y (Y,N,A,R)
  GRANT access ON TABLESPACE . . . Y (Y,N,A,R)
  GRANT access ON TABLE  . . . . . Y (Y,N,A,R)
  GRANT access ON VIEW . . . . . . Y (Y,N,A,R)
  GRANT use OF STORAGE GROUP . . . Y (Y,N,A,R)
  ALTER TABLE ADD FOREIGN KEY  . . Y (Y,N,D)
  LABEL ON . . . . . . . . . . . . Y (Y,N)
  COMMENT ON . . . . . . . . . . . Y (Y,N)
  ALTER TABLE ACTIVATE CONTROL . . Y (Y,N)

Other GEN options:
  New TS storage group  . . .            >
  New IX storage group  . . .            >
  New database  . . . . . . .
  New schema of objects . . .            >
  New grantor . . . . . . . .            >

  Use Masking . . . . . . . . . . . N (Y,N)
  IDENTITY START value  . . . . . . ORIGINAL (Original,Computed)
  Run SQLID . . . . . . . . . . .            (Blank, a SQLID, <NONE>)
  Retain GENERATED ALWAYS:
    For ROWID . . . . . . . . . . . N (Y,N)
    For ROW CHANGE TIMESTAMP  . . . N (Y,N)
```

Figure 225. Generate SQL from DB2 catalog (ADBP8MG)

Step 2. Generate batch jobs

After you have specified the objects to migrate information, generate batch jobs.

Before you begin

Add the objects that you want migrated to the list.

About this task

To generate batch jobs:

Procedure

1. Issue the MIG primary command to start generating jobs for the migration. The Migrate Parameters panel is displayed, as shown in the following figure.

```
ADB28M ------------------ DB2X Migrate Parameters ------------------ 09:58
Option ===>

Please specify the following for DB2 Admin Migrate:     DB2 System: DSNA
                                                        DB2 SQL ID: ISTJE
                                                              More:    +
Worklist name . . . . . . .             (also used as middle qualifier in DSNs)

Data set information:
  PDS for jobs  . . . . . .   MYMIGR.JCL
  Prefix for datasets . . .   ISTJE

Target system parameters:
  DB2 subsystem id (SSID) .  DB2X        DB2 release  . . . . : 1101
  Target system node name .               Submit job at local. : NO  (Yes/No)
  DB2 sample pgm load lib .   DBS.DSN110.RUNLIB.LOAD
  Target JCL job data sets for Admin and DB2
    Use customization settings for Admin libs . . . . . NO  (Yes/No)
    Use customization settings for the following libs . NO  (Yes/No)
      DB2 Admin APF library . .
      DB2 exit library  . . . .
      DB2 load library  . . .   SYS1.DSNDB2X.SDSNLOAD
  Catalog statistics options:
    Catalog qualifier . . . . . HI         >  (default SYSIBM)
    Statistics tables . . . . . SELECT        (All or Select. Default is All)

Migrate options:
  Generate MIG jobs in batch  . . .   NO      (Yes/No)
  Generate work stmt list . . . . .   NO      (Yes/No)
  Combine job steps . . . . . . . .   YES     (Yes/No, Yes if HPU Unload)
    Member prefix for combined jobs   ADBMG   (default ADBMG )
  Scope of migrate:
    DDL . . . . . . . . . . . . . .   N       (Yes/No)
    Data  . . . . . . . . . . . . .   N       (Yes/No)
    Catalog statistics  . . . . . .   N       (Yes/No)
  DROP on target before CREATE  . .   NO      (Yes/No,No if scope DDL is NO)
  Unload method . . . . . . . . . .   U       (U - Unload, H - HPU, C - Cross)
  Parallel utilities  . . . . . . .   NO      (Yes/No)

Optional steps after reload:
  Run CHECK DATA  . . . . . . . . . : NO      (Yes/No)
  Run RUNSTATS  . . . . . . . . . . : NO      (Yes/No)
  Run IMAGE COPY  . . . . . . . . . : NO      (Yes/No)
  Run REBIND  . . . . . . . . . . . : NO      (Yes/No)

Utility control options:
  Generate template statements  . . :         (Yes/No)
  Use customized utility options  . :         (Yes/No)

BP  - Change batch job parameters
TU  - Specify template usage
UO  - Customize utility options
GEN - GEN options
```

Figure 226. Migrate Parameters panel (ADB28M)

2. Specify the following information on the Migrate Parameters panel:
   - The PDS where the generated jobs are to be stored
   - Data set information
   - Target system parameters
   - Migrate options
   - Optional jobs to be run after the reload
   - Utility control options
   - Gen options

   You can modify options without leaving the MIG area. Refer to the online help for detailed information about the fields in the panel.

   If you specify to have the migrate jobs generated in batch, DB2 Admin creates a work data set (MIGVARS) that stores the parameter information specified on

the panel and the necessary ISPF tables to use as input for the generation of the migrate source and target JCL. Similar to the other migrate work data sets that are used, you can use the Prefix for datasets field and the Worklist name field to change the default qualifier values that are used for the MIGVAR data set.

If you choose Unload as the unload method and parallel utility processing and do not provide your own UNLDDN template, the default template ASYREC6 with variable &PART or &PA in the ADB2UCUS skeleton is used as the template for the unload data set. When &PART or &PA is specified, DB2 Admin replaces the variable with 00001 up to the maximum partition number of the associated object. The total length of the values for &PREFIX and &LEVEL must not exceed 12 bytes.

If you choose to migrate only the data, then use the LOAD utility option REPLACE and RESUME to control how the data is loaded into the target system. You can customize the LOAD utility options, as needed, using the UO - Customize utility option command. Set the option **Use customized utility options** to YES. If customized utility options are not used and **Use customized utility options** is set to NO, then Migrate uses the default REPLACE and RESUME option generated by the DB2 UNLOAD utility or the High Performance Unload (HPU) utility. You can customize the HPU utility by entering the HPU option on the Change Utility Options panel, as described in "Using DB2 High Performance Unload" on page 427. Other considerations for the REPLACE and RESUME option are as follows:

- DB2 does not allow using LOAD REPLACE on certain types of tables. When LOAD REPLACE is not allowed but is specified, the REPLACE option is converted to RESUME YES.
- If not all the tables under a multi-table table space are selected for migration on the source system, the REPLACE option for LOAD utility, if specified, is converted to RESUME YES.
- If all the tables under a multi-table table space are selected for migration on the source system, the REPLACE option for LOAD utility, if specified, is used. Any additional tables under the table space on the target system remain empty after migration.

Note: The CHECK, RUNSTATS, and COPY requests are not generated for implicitly created table spaces.

The REBIND option generates rebinds of the source packages for the target system.

Restriction: If you specify Yes for the DROP on target before CREATE field, any RESTRICT ON DROP conditions for tables are not considered. If a table has RESTRICT ON DROP, you will need to remove it for the DROP statement to complete successfully.

3. Press Enter. DB2 Admin starts to generate the jobs required for migration.

   The panels that are displayed and the action to take next depends on whether you chose to generate the migrate jobs online or in batch.

4. Specify whether to generate the migrate jobs online or in batch.

   - If you choose to generate the migrate jobs online, review the messages that are displayed in the Migrate Progress pop-up panel. These messages provide information about the status of building the jobs.

     When DB2 Admin finishes generating the jobs, it invokes an ISPF Edit session. An example of this edit screen is shown in the following figure.

Press F3 to exit the edit session.

```
  Menu  Functions  Utilities  Help
--------------------------------------------------------------------------------
EDIT     ISTJE.MIGDSN85.JCL                               Row 00001 of 00011
Command ===>                                                Scroll ===> PAGE
   Name      Prompt        Size   Created        Changed           ID
. SST1RE                     60   2007/11/25   2007/11/25 00:55:00   ISTJE
. SST2UL1                    64   2007/11/25   2007/11/25 00:55:00   ISTJE
. SST3CH                     34   2007/11/25   2007/11/25 00:55:00   ISTJE
. SST4XF                     19   2007/11/25   2007/11/25 00:55:00   ISTJE
. SST5DE                     29   2007/11/25   2007/11/25 00:55:00   ISTJE
. TST1CR                     23   2007/11/25   2007/11/25 00:55:00   ISTJE
. TST2RL                     96   2007/11/25   2007/11/25 00:55:00   ISTJE
. TST3CK                     35   2007/11/25   2007/11/25 00:55:00   ISTJE
. TST4RS1                    23   2007/11/25   2007/11/25 00:55:00   ISTJE
. TST5IC                     58   2007/11/25   2001/11/25 00:55:00   ISTJE
. TST7DE                     29   2007/11/25   2007/11/25 00:55:00   ISTJE
  **End**
```

Figure 227. Sample migrate edit panel

- If you choose to generate the migrate jobs in batch, submit the job that is displayed in the ISPF Edit session that is invoked. This batch job generates the jobs that are required for migration.

  The member name for the batch job is either
  – `<Member prefix for combined jobs>.S0`, if you chose to combine the job steps
  – `SST0BAT`, if you chose not to combine the job steps

  An example of the edit screen is shown in the following figure.

```
  Menu  Functions  Utilities  Help
--------------------------------------------------------------------------------
EDIT     ISTJE.MIGDSN85.JCL                               Row 00001 of 00011
Command ===>                                                Scroll ===> PAGE
   Name      Prompt        Size   Created        Changed           ID
. ADBGS0                     83   2007/11/25   2007/11/25 00:55:00   ISTJE
  **End**
```

Figure 228. Sample of job edit panel for generating the migrate jobs in batch

Results

You are ready to review, edit, and run the generated migrate jobs.

Step 3. Run the batch jobs

After you have generated the batch jobs, you can run them.

About this task

To run the batch jobs:

Procedure

1. Review the following source system jobs and submit them in the sequence shown.
   a. SST1RE - Performs reverse engineering
   b. SST2UL*n* - Unloads data; *n* is an integer. If you are migrating many table spaces, multiple unload jobs might be created.

c. SST3CH - Changes unload control data sets

2. Run the batch jobs.

   - If you combined the job steps, these jobs are located in the group xxxxS1. Run the first group having the name of xxxxS1 on the source system.

   - If you specified the current system node name as the name of the target system node name, the source and target systems are the same. You will run all the generated jobs on the same system. You will skip Step 4 to transfer the batch jobs that begin with T (or group xxxxT1 if you combined the job steps) to the target system.

   - If you request that a work statement list be generated and are running in local mode (that is, not connected to a remote subsystem), run the xxxSn job in sequence to extract the DDL, unload the data, change the load control statements, and write the work statement list.

   - If you are running in DRDA mode (that is, connected to a remote system), run the xxxSn job first to unload the data on the remote (source) system. After the xxxSn jobs are complete and the data sets contain the unloaded data and the load control statements are transferred from the remote system to the local system, run the xxxLn job to extract the DDL, change the load control statements, and write the work statement list.

     Usually, only one xxxSn job exists to unload the data. However, if many table spaces require unloading, multiple xxxSn jobs are generated. The final xxxSn job on the remote system specifies the data set names that need to be transferred to the local system for creating the work statement list. The n in the xxxLn job is one greater than the n in the last xxxSn job.

Step 4. Optional: Transfer the jobs/work statement list and data to the target system

After you have run the batch jobs, perform this step only if the source and target systems are different; that is, the node names for the source and target systems are not the same.

About this task

If the source and target database systems are on separate machines, it might be necessary to transfer the information electronically or by using a portable medium, such as a tape.

To transfer the jobs/work statement list and data to the target system:

Procedure

Run the following jobs in the sequence shown:
- SST4XF - Information about the data sets that needs to be transferred
- SST5DE - Delete data sets on source system

Results

If you combined job steps, these jobs are located in the group xxxxSE. Run the second group having the name of xxxxSE on the source system after all jobs in the first group are complete.

If your source and target DB2 subsystems are on the same machine, do not run the delete data sets on the source system job (SST5DE or xxxxSE) until you run all of the jobs for the target system.

If you request that a work statement list be generated, the job name xxxSE is used (when not connected to a remote system). Otherwise, the name xxxLE is used. This job specifies the data set names with the work statement list that is required to be transferred to the target system, along with a job step to delete the data sets. Do not run the step to delete the data sets if you are using the work statement list.

Step 5. Run the batch define, reload, and optional jobs

After you have run the batch jobs or transferred the jobs/work statement list and data to the target system, run the batch define, reload, and optional jobs.

About this task

To run the batch define, reload, and optional jobs:

Procedure

Review the following target system jobs and submit them in the following sequence:
1. TST1CR - Creates objects and changes the catalog statistics (updates, inserts, and deletes) on target system.
2. TST2RLn - Reloads data; n is an integer. If many table spaces are being reloaded, multiple reload jobs can be created.
3. TST3CK - Performs CHECK DATA (optional).
4. TST4RS - Runs RUNSTATSs (optional).
5. TST5IC - Performs an image copy (optional).
6. TST6RB - Rebinds (optional).
7. TST7DE - Deletes data sets on target system.

Results

If you combined job steps, these jobs are located in the group xxxxT1. Run group xxxxT1 on the target system. If you performed "Step 4. Optional: Transfer the jobs/work statement list and data to the target system" on page 344 to transfer the jobs and data to the target system, ensure that all the jobs from group xxxxSE are complete before running group xxxxT1.

If you specified the current system node name as the name of the target system node name, the source and target systems are the same. Therefore, you will run these jobs for Step 5 on the same system as the jobs that you ran for the source system in "Step 3. Run the batch jobs" on page 343.

Work data sets used by the DB2 Admin Migrate function

The DB2 Admin Migrate function creates and uses data sets.

The following figure shows the data sets that the DB2 Admin Migrate function creates and uses.

Table 12. Work data sets for DB2 Admin Migrate

Default data set name	Description	Template keyword
prefix.worklist.DDL	DDL and DML that is constructed from the catalog	MISQL
prefix.worklist.DDDL	DROP statements for drop objects	MISDROP
prefix.worklist.COL	Identity column information	MICOL
prefix.worklist.CMD	Rebind output	MIGCMD
prefix.worklist.MIGVARS	Partitioned data set for ISPF tables that are required for generating the MIG jobs in batch	MIGSHVR
prefix.worklist.ADB28W1U	Work statement list data set	MIUCONV
prefix.worklist.ADB28W3U	Work statement list data set	MIUOTHR
prefix.worklist.ADB28WDD	Work statement list elements	MI2WDD
prefix.worklist.ADB28W2T	Input data set for the merge program	MIMLSIN
prefix.worklist.ADB28W2U	Intermediate data set used by the merge program	MIMLSOT

The DB2 Admin Migrate function also uses data sets for the unloaded data, load control statements, and converted load control statements. The naming convention for the data sets differ depending on whether the DB2 UNLOAD utility or DB2 High Performance Unload (HPU) is used to unload the data.

The following figure shows the data sets for migrations with DB2 UNLOAD.

Table 13. Work data sets for DB2 Admin Migrate with DB2 UNLOAD

Default data set name	Description	Template keyword
prefix.worklist.CNT.S*n*	Load utility control statements, where S*n* is a string assigned to the object by DB2 Admin, with *n* beginning with 1	PUNCHDDN[1]
prefix.worklist.ULD	Data sets for unloaded data	UNLDDN[2]
prefix.worklist.CNC.S*n*	Converted load utility control statements, where S*n* is a string assigned to the object by DB2 Admin, with *n* beginning with 1	MICTLOV (for table spaces MICTLOU (for tables)

Note:

1. A utility template. A template statement is not generated in the JCL. DB2 Admin uses the utility template to generate regular JCL to perform the unload.
2. A utility template. A template statement is generated in the JCL. When you use your own copy of utility template UNLDDN, DB2 Admin does not delete any of the data sets that are created by the template after they are used. You must delete them. Also, the transfer data set list in jobs SST4XF and xxxxSE do not include the data set names, and you must transfer them.

Image copy uses the regular utility template.

The following figure shows the data sets for migration with HPU.

Table 14. Work data sets for DB2 Admin Migrate with HPU

Default data set name	Description	Template keyword
prefix.worklist.CNT.T*n*	Load utility control statements, where T*n* is a string assigned to the object by DB2 Admin, with *n* beginning with 1	MICTLIU
prefix.worklist.ULD.T*n*	Unload data sets for a non-partitioned object, where T*n* is a string assigned to the object by DB2 Admin, with *n* beginning with 1	MIDTVNP
prefix.worklist.ULD.T*n*.P*m*	Unload data sets for a partitioned object, where T*n* is a string assigned to the object by DB2 Admin, with *n* beginning with 1, and P*m* is a string that identifies the object's partition number, with *m* beginning with 0001	MIDATVP
prefix.worklist.CNC.T*n*	Converted load utility control statements, where T*n* is a string assigned to the object by DB2 Admin, with *n* beginning with 1	MICTLOU

The relationship between the table name and T*n* and the relationship between the table space name and the S*n* are listed as comments in the front part of the generated job or work statement list.

DB2 Admin Migrate deletes these data sets when they are no longer needed.

Creating naming conventions for work data sets that are created by the DB2 Migrate function

You can use templates to create your own naming conventions for the work data sets that are created by the DB2 Admin Migrate function.

About this task

To use templates to create naming conventions for the work data sets that are created by the DB2 Admin Migrate function:

Procedure

1. Specify YES in the **Generate template statements** field on the Migrate Parameters panel.
2. Use the TU primary command on the Migrate Parameters panel (or Option 5 on the Administration Menu panel) to manage the templates for the work data sets. You can use the TU primary command on the Alter Tablespace Redefine - JCL panel (or Option 5 on the Administration Menu panel) to manage the templates for the work data sets.

   The valid variables that can be specified when constructing the data set name pattern for a template for a migrate work data set include:

   - The following functional variables:

     &ADB28PRE.
     Prefix for data sets specified on the Migrate Parameters panel (ADB28M)

Chapter 14. Migrating DB2 objects, data, views, and catalog statistics   **347**

&DB2SYS.
>The DB2 subsystem id

&WORKLIST.
>Worklist name specified on the Migrate Parameters panel (ADB28M)

- The following variables that are supported for normal DB2 utility template processing:

&DB.   Database name

&TS.   Table space name

&PART.
>The value is ALL with these exceptions:
>- For template UNLDDN, DB2 z/OS resolves the variable to a 5-byte string (*nnnnn*) that represents the partition number. For a non-partitioned object, the value of the string is '00000'. For a partitioned object, the value of the string is '00001', '00002', and so on.
>- For template MIDATVP with parallel processing specified, DB2 Admin resolves the variable to a 4-byte string (*nnnn*) that represents the partition number. The value of the string is '0001', '0002', and so on.

&USERID.
>Batch user ID

&DATE.
>YYYYDD

&TIME.
>HHMMSS

&JDATE.
>YYYYDDD

&YEAR.
>YYYY

&MONTH.
>MM

&DAY.
>DD

&JDAY.
>DDD

&HOUR.
>HH

&MINUTE.
>MM

&SECOND.
>SS

>This list of variables is a subset of the variables that are supported for normal DB2 utility template processing. The other variables tht are supported for normal DB2 utility processing are not valid.

This list of variables is a subset of the variables that are supported for normal DB2 utility template processing. The other variables that are supported for normal DB2 utility processing are not valid.

Restriction: The following restrictions apply when specifying variables:

- For the data set names to which DB2 Admin appends S*n*, T*n* or T*n*.P*mmmm* , the number *n* starts with 1 and ends with the number of objects that you want to migrate. The total length of a data set name should not exceed 44 bytes

- The only variables that can be specified for UNLDDN (used when using DB2 UNLOAD) are &DB., &TS., &USERID., and &PART..

  A work list name can be a very important part of the data set name when migrating objects. To specify a work list name as part of UNLDDN template data set name to maintain a consistent naming convention with other data set names, explicitly specify a value in the template instead of using the variable &WORKLIST, which cannot be specified.

- The only variables that can be specified for MIDTVNP and MIDATVP (used when using HPU) are &DB., &TS., &USERID., &WORKLIST., and &PART.. If you specify &PART. for MIDATVP, you must specify it as the last part of the name (for example, &USERID..&TS..ULDULD.P&PART.); otherwise, parallelism will not be performed.

Chapter 15. Using work statement lists

DB2 Admin work statement lists (WSLs) allow you to create and maintain a set of operations that you can run online or in batch mode.

You can run the entire set of operations, rerun sets of operations or capture a set of operations that are created on one subsystem and use those operations on another subsystem.

Topics:

Work statement lists

A *work statement list*, or WSL, is a collection of one or more tasks that perform basic operations.

In general, the statements in a WSL are standard statements or commands that you would normally code to perform a task. Entries in a WSL can include items in any of the following categories:

- SQL statements:
    - Data definitions, such as CREATE, DROP, ALTER, and RENAME
    - Authorization changes, such as GRANT and REVOKE
    - Data manipulation changes, such as INSERT, UPDATE, and DELETE
- DSN commands: such as BIND, REBIND, FREE, and RUN
- DB2 commands: such as START, STOP, ALTER, and SET
- Utilities statements
- REXX and CLIST statements
- DB2 Admin instructions, which follow a product-specific syntax for performing certain complex operations.

Certain functions in DB2 Admin support or produce input and output statements that are used by DB2 or by DB2 Admin. IBM might provide an alternate statement or alternate form for clauses on statements, and might identify one as the preferred syntax, while still supporting the alternate form.

DB2 Admin might use preferred or alternate forms of syntax. If the statement produced is accepted by the products or by DB2, the statement is considered valid. Where it is necessary to produce an accepted statement, the products convert to newer syntax. However, the products might retain older syntax even if DB2 considers the newer syntax the preferred syntax. This might be the case even if no possible use of the older syntax is needed. The use of older syntax might persist until IBM deems it is no longer is supported in any product form.

Creating work statement lists

You can create WSLs in several different ways.

You can create WSLs in one of the following ways:
- By using DB2 Admin basic functions: definition SQL, authorization SQL, update SQL, DSN commands, and DB2 commands
- By using output from the DB2 Admin Reverse Engineering function
- By using the DB2 Admin Alter Table Columns function
- By using one of the DB2 Admin utilities panels
- By coding a WSL manually
- By cloning an existing WSL member

Using DB2 Admin basic functions to create WSLs

You can use DB2 Admin basic functions to create WSLs.

To create WSLs using the following DB2 Admin basic functions, activate prompting using the PROMPT primary command. REXX and CLIST statements are not activated via PROMPT. There is no comparable method.
- Definition SQL (CREATE, DROP, ALTER, and RENAME)
- Authorization SQL (GRANT and REVOKE)
- Update SQL (INSERT, UPDATE, and DELETE)
- DSN commands (BIND, REBIND, FREE, and RUN)
- DB2 commands (START, STOP, ALTER, and SET)
- REXX and CLIST statements

Recommendation: Use the PROMPT Options panel to activate the Prompt facility. The Prompt facility allows you, on a statement type level, to specify whether prompting is active for the statement type. Once activated, you are prompted before DB2 attempts to execute the statement type. When prompted, you can choose to do one of the following:
- Execute the statement.
- Edit the statement.
- Create a batch job with the statement.
- Add the statement to a WSL. Specify the WSL library and member name.

Using Reverse Engineering to create WSLs

You can create a WSL with Reverse Engineering using either the GEN line command (or primary command) or the DDL line command.

Directing the output of the GEN command to a WSL:
About this task

To direct the output of the GEN command to a WSL:

Procedure
1. On the ADB2GEN panel, specify a `Y` in the **Add to work stmt list** field.
2. Specify the WSL library and member name when you are prompted.

What to do next

If the WSL name already exists, you can choose to add the GEN output to the end of the current contents of that WSL or to replace the current contents of the WSL with the GEN output.

Directing the output of DDL to a WSL:
About this task

To direct the output of DDL to a WSL:

Procedure
1. Set PROMPT ON.
2. Specify Y in the **Execute the generated SQL** field.
3. Press PF3 or the End key.
4. Select option 4 to add the statement to the work statement list.

Using DB2 Admin Alter table columns to create WSLs

The DB2 Admin Alter (ALT) panel is used to specify the names and options for DB2 Admin Alter.

On this panel, you can elect to use a WSL.

After entering information in the fields, you are prompted to specify the WSL library and member name. If the WSL name already exists, you can choose to add the ALT output to the end of the current contents of that WSL or to replace the current contents of the WSL with theALT output. The next panel displays the JCL that you must run to populate the WSL.

Using DB2 Object Comparison Tool Apply tasks to create WSLs

If you use DB2 Object Comparison Tool, you can use the Generate Compare Jobs panel (option 5 on the DB2 Object Comparison Tool menu) to add Apply tasks to a WSL.

After entering information in the fields, you are prompted to specify the WSL library and member name. If the WSL name already exists, you can choose to add the Apply tasks to the end of the current contents of that WSL or to replace the current contents of the WSL with the Apply tasks. The next panel displays the JCL that you must run to populate the WSL.

Using the DB2 Admin Utilities panels to create WSLs

You can use the DB2 Admin Utilities panels to create WSLs by specifying that utility statements be placed into a WSL and specifying the WSL library and member name.

On the following panels, you can specify that utility statements be placed into a WSL:
- Table Utilities panel (ADB2UT)
- Table Space Utilities panel (ADB2US)
- Index Utilities panel (ADB2UX)
- Storage Group Utilities panel (ADB2UG)
- Create Index Utilities panel (ADB26CXU)
- LISTDEF panel (ADB25LU)

After entering information in the fields, you are prompted to specify the WSL library and member name. If the WSL name already exists, you can choose to add the utility statements to the end of the current contents of that WSL or to replace the current contents of the WSL with the utility statements.

Coding a WSL manually

You can edit a WSL to enter work statements directly.

The following statement types can be added to a WSL:
- Comment statements
- Definition SQL statements
- Authorization SQL statements
- Update SQL statements
- DB2 commands
- DSN commands
- DB2 utility statements
- DB2 Admin statements
- REXX and CLIST statements

Using delimited identifiers when creating work statement lists

When creating WSLs, you can use quotation marks with delimited identifiers in a statement.

If you clone a WSL that includes a statement containing delimited identifiers, DB2 Admin removes the quotation marks from the identifier if it does not require delimiters.

A WSL contains the following DDL:

```
DDL  CREATE SYNONYM "PROJSYN" FOR "DBA282"."PROJ"
```

The cloned result does not contain the quotation marks:

```
COM  -- Created by DBA282 on 2002/07/23 at 15:23 by cloning of
COM  -- source work stmt list  RESULT  from library  WSL.DATA
DDL  CREATE SYNONYM PROJSYN FOR DBA282.PROJ
```

Where work statement lists are stored

Work statement lists are stored in ISPF tables in a data set that you specify.

They can be accessed by other users and are protected by RACF. By storing WSLs in ISPF tables, they can easily be moved to other systems or installations. A WSL can be created on one system and changed or executed on another system. The following scenarios are possible:
- Local use only: Generate the WSL on subsystem DB2-1. Clone the WSL many times with different owners and names for the objects. Execute the WSL on DB2-1.
- Local customization and remote execution: Generate the WSL on subsystem DB2-1. Clone the WSL many times with different owners and names for the objects. Send the WSL to subsystem DB2-2. Execute the WSL on DB2-2.
- Remote customization and execution: Generate the WSL on subsystem DB2-1. Send the WSL to DB2-2. Clone the WSL many times with different owners and names for the objects. Execute the WSL on DB2-2.

Restriction: Do not use the DDL line command to generate the SQL for a specific WSL. You can manually edit an existing WSL using the specified option provided on the WSL panel.

How running a work statement list works

You run a WSL by entering a line command on the Work Statement Library List panel (ADB2W1).

You can run a WSL either in batch (the R line command) or online (the O line command).

When you use the R line command to run a WSL in batch, one or more jobs are created. Each job includes a step to run the Batch Restart Program ADBTEP2 and the job's set of input instructions (batch statement list) for ADBTEP2.

When you use the O line command to run a WSL online, ADBTEP2 is run online and all input instructions are processed sequentially.

When you run a WSL in batch, DB2 Admin generates multiple jobs when it encounters the PARALLEL command in the WSL. DB2 Admin generates the following job names:

`<prefix><m><seqnumber>`

<prefix>
   A specified prefix. The prefix can be from 4 to 6 characters, depending on the number of parallel jobs.

<m> The first character in the word following the PARALLEL command. For example, U for UNLOAD; R for RELOAD.

<seqnumber>
   The generated sequence number. The sequence number can be from 1 to 3 characters (*n* to *nnn*), depending on the number of parallel jobs:

n	For 1 to 9 parallel jobs
nn	For 10 to 99 parallel jobs
nnn	For more than 99 parallel jobs

The maximum length of a job name is 8 characters.

You can restart failed work statement list jobs by re-issuing the R or O line command on the Work Statement Library List panel (ADB2W1). If the WSL contains PARALLEL processing capability, the WSL must be restarted in the same mode that it was originally run (either online or batch). A failed parallel process that was originally submitted as a batch job cannot be restarted in online mode, and vice versa.

Managing work statement lists

You can use DB2 Admin to manage WSLs.

DB2 Admin enables you to perform the following tasks:
- Show the content of a WSL
- Analyze the content of a WSL and assess the impact of running it
- Edit a WSL statement
- Generate a job to run the WSL in batch

- Run a work statement list and view the automatically generated Load Summary Report
- Resubmit a work statement list that was run by another user that did not complete successfully
- Delete a WSL from the library
- Copy a WSL and append it to another WSL
- Clone an existing WSL to run on a different DB2 subsystem or against DB2 objects of different naming schemes
- Add output from storage group, table space, table, and index utilities to a WSL
- Add ALTER TABLE (ALT) requests to a WSL (you can alter multiple tables by appending several requests on one WSL)

To manage WSLs, select option W on the Administration Menu panel to display the Manage Work Statement Lists panel, as shown in the following figure. This panel allows you to either view the entire WSL library or just a single WSL. You can also issue the WSL primary command from any DB2 Admin panel to display the Manage Work Statement Lists panel.

```
DB2 Admin --------------- DB2X Manage Work Statement Lists -------------
Option ===>

  1 - Show work statement list library              DB2 System: DB2X
  2 - Show work statement list                      DB2 SQL ID: ISTJE

 Work stmt list dsn  ===> TEST.WL
 Work stmt list name ===> SI
```

Figure 229. Manage Work Statement Lists panel (ADB2W)

Use this panel to manage an entire WSL library or to manage a single WSL.

Recommendation: When working with a WSL that has been generated to implement changes that are being made through Change Management, do not run the WSL from the Work Statement List Library panel (ADB2W1). Instead, use the RN command on the CM - Changes panel (ADB2C11) to run the change, which causes the WSL to be run. Use the RN command because any change that you want to track through Change Management must be made through Change Management. It is also recommended that you do not use the line commands on the Work Statement List Library panel to edit, delete, copy, append, or clone the WSL.

Viewing a WSL library

You can view and manage an entire WSL library.

About this task

To view and manage an entire WSL library:

Procedure

1. In the WSL **dsn** field, specify the data set name of the ISPF library that contains the WSLs.
2. Select option 1 on the Manage WSL panel.
3. Press Enter.

   The Work Statement List Library panel is displayed, as shown in the following figure.

```
 ADB2W1 in ---------------- Work Statement List Library ------ Row 1 to 6 of 6
Command ===>                                               Scroll ===> CSR

Commands: OPTIONS
Line commands:
 S - Show  R - Run (batch)   D - Delete  C - Copy  A - Append  Q - Clone
 I - Interpret  V - Validate  E - Edit  B - Checkpoint

Work Statement List: WSL.LIST

Sel Name     Created    Changed           ID       Restart
    *        *          *                 *        *
--- -------- ---------- ---------------- -------- --------
    C0000023 2013/04/05 2013/04/05 09:15 VNDDHG
    DG29608  2013/04/05 2013/04/05 14:17 VNDDHG
    D1026676 2013/03/27 2013/03/27 10:03 VNDDHG
    D25359   2013/03/27 2013/03/27 10:47 VNDDHG
    RE026676 2013/03/18 2013/03/18 14:04 VNDDHG
    S28654   2013/02/28 2013/03/28 12:38 VNDDHG   Y
```

Figure 230. Work Statement List Library panel (ADB2W1)

This panel shows the contents of the WSL library, with each list on a separate line.

Use the following line commands to manage WSLs within a WSL library:

S   Show the WSL.

R   Run in one or more batch jobs.

D   Delete the WSL from the library.

C   Copy this WSL and append it to the WSL where the A line command has been or will be specified.

A   Append the WSL to this member from where the C line command has been or will be specified.

Q   Clone the existing WSL member for use on other DB2 subsystems.

I   Interpret the WSL.

V   Validate the syntax of the SQL statements in the WSL and provide an impact analysis of the objects that would be affected by running the WSL.

E   Invokes ISPF EDIT so you can edit the WSL. Upon exiting from EDIT mode, the original WSL is updated.

When editing the WSL, you must end each statement with the current delimiter character. The delimiter character is a semicolon unless a -#SET TERMINATOR functional comment precedes the statement.

Tip: To perform a search for a string in the WSL, invoke the EDIT command to display all of the statements, then use FIND to search for a specific text string.

O   Run the WSL online.

When you run a WSL online, certain program or utilities that are intended to be run in batch might issue messages to the terminal. Make note of these messages, and press Enter to clear the messages.

Cloning work statement lists

You can clone work statement lists.

About this task

To clone a WSL:

Procedure

1. Select option 1 on the Manage Work Statement Lists panel.
2. On the Work Statement List Library panel (ADB2W1), issue the Q line command on a WSL that you want to clone. The Clone Work Statement List panel is displayed, as shown in the following figure.

```
DB2 Admin ------- Clone Work Statement List ---------------------------------
Command ===>                        Scroll ===> PAGE

 Input work stmt list information:                     DB2 System: DB2X
   Work stmt list . . . : SRCEWSL                      DB2 SQL ID: ISTJE
   from library . . . . : WORKLIST.LIB
                                                           More:     +
 Output work stmt list information:
   Library (PDS name) . : WORKLIST.LIB2
   Work stmt list . . . : UNION2    (will be new PDS member)

 Execution mode . . . . : BATCH     (BATCH or TSO)
   PDS for jobs  . . .. : ISTJE10
   PDS member . . . . . : WORKLISTS
   Unit type  . . . . . : SYSDA

 Use Masking. . . . . . : NO       (Yes/No)
   Apply masking to data set names. . :         (Yes/No=default)

 Use local DB2 catalog information to replace: (Yes/No)
   Authorizations . . . . . . . . . . :
   Partitioning attributes  . . . . . :
   Table space and index attributes . :

 Additional parameters:
   Message output file  : 'ISTJE.CLONE.SYSPRINT.SRCEWSL'
```

Figure 231. Clone Work Statement List panel (ADB2W1Q)

3. In the **Input work statement list information** fields, the WSL that you selected and library in which it is stored is displayed. If necessary, change these names.
4. In the **Output work stmt list information** fields, specify a name for the new WSL and a name of a library in which to place it.

   Library (PDS name)
   The name of a library in which to place the new WSL as a new PDS member. Use standard TSO format for this name. If this PDS does not exist, DB2 Admin creates and catalogs this PDS with a default size of 1 cylinder, record length 80, and fixed-block with a block size of 6160.

   Work stmt list
   The name of the new (cloned) WSL. DB2 Admin creates a new PDS member using this name in the PDS/library that you specify. If a PDS member by this name already exists in that library, the PDS member is not replaced, and the cloning attempt fails.

   WSLs are stored as ISPF tables, which are subject to the ISPF restriction that requires currently active tables to have different names, even when the tables are from different PDS/libraries. Therefore, the cloned WSL that you create and its source WSL must have different names.

5. Select a mode for running the cloning job, either batch or TSO. If you select batch mode, specify values for the "PDS for jobs" and "PDS member" fields. If necessary, change the default unit type.

6. Specify `Yes` or `No` in the **Use Masking** field. If you specify Yes, the Specify Mask panel is displayed, and you can specify the mask to use and edit the mask definition before you begin the cloning process.

   Masking enables you to change the names of the DB2 objects, owners, and schemas that are referenced in the original WSL for use in the new (cloned) WSL. Masking also enables you to specify overwrite values for several table space and index space attributes. Masking is often useful when the new (cloned) WSL is to be used on a different DB2 subsystem or in a different database.

   Remember: Cloning always leaves the original WSL unchanged.

   If the target DB2 subsystem exists on a remote site, you can use the standard TSO services to send the newly cloned WSL to that remote site. Or, you can send the original WSL to that remote site first, and complete the cloning on that remote site.

   Tip: The fields to specify overwrite values for table space or index space attributes are no longer available on this panel because masking is changed to include the support to specify overwrite values for PRIQTY and SECQTY. When you edit the mask while under the control of the DB2 Admin masking macro, you can import your old overwrite values by using the command, COPY '*overwrites_data_set_name*' after .ZL, where '*overwrites_data_set_name* is the name of your old overwrites data set. The COPY command appends the contents of the specified overwrites data set to the mask contents.

   The following field is also available for specifying values:

   Apply masking to data set names
   Specifying `Yes` in this field causes name masking to be applied to data set names. Name masking is useful when DB2 Admin generates data set names with qualifiers that are based on database object names. This field only affects the following statements: TSO ALLOCATE, ADM TSODELETE, UTL TEMPLATE, and UTL UTLFROM(admin).

7. Decide whether to override the existing authorizations, partitioning, and table space and index attributes.

   Authorizations
   Overrides authorization to objects that are created by the WSL with authorization records (grants) from the local DB2 catalog. Grants to objects that are not created by the WSL are not overridden.

   Partitioning attributes
   Overrides characteristics of partitioned tables spaces and indexes in the local DB2 catalog. Objects that are not partitioned in the local DB2 catalog are not affected. The list of columns that comprise the index key is not overridden. This index property is always taken from the WSL statement.

   Restriction: Certain conditions make it impossible to override partitioning. For example, it is unsafe to change partitioning attributes if the list of index columns in the WSL statement is not a strict extension of the list of index columns found in the local DB2 catalog. In this case (for an index on a table), no partitioning attributes are overridden.

Table space and index attributes

For the CREATE TABLESPACE/INDEX statements, in the newly cloned WSL, you can replace the primary and secondary quantity values specified in these statements with the values from the local DB2 catalog tables (SYSTABLEPART and SYSINDEXPART) where cloning is requested. If the masking feature is used, the masking to change DB2 object names and owners is performed first, then any overwrite values that are specified for PRIQTY and SECQTY, if any, are performed using the new table space or index names.

8. Optional: Overwrite the attributes for table spaces and index spaces. Specify whether to edit the data set.
9. Specify an output message data set in the **Message output file** field.
10. Press Enter to complete the cloning process.

Results

While using the Clone Work Statement List panel, you can browse the message data set for a cloned WSL by specifying the appropriate message output file (if it was changed from the default) and issuing the M primary command.

Cloning can be performed on a WSL containing any valid commands and valid SQL statements.

Viewing a WSL

You can view and manage a single WSL.

About this task

To view and manage a single WSL:

Procedure

1. In the Manage WSL panel, specify the data set name of the ISPF library that contains the WSL and the name of the WSL. If the WSL does not exist, DB2 Admin creates it for you.
2. Select option 2 and press Enter. The Show Work Statement List: CREATE panel is displayed, as shown in the following figure.

```
DB2 Admin ----------- Show Work Statement List: CREATE ------------ Row 1 of 4
Command ===>                                               Scroll ===> Page

Line commands:
 D - Delete  I - Insert  E - Edit  C - Copy  M - Move  A - After  B - Before
 R - Repeat

Select Type Statement
       *    *
------ ---- ----------------------------------------------------------------
       DDL  CREATE DATABASE "YYYY2"    STOGROUP "ISTJEG"
       DDL  CREATE TABLESPACE "YYYY2S"    IN "YYYY2"     USING STOGROUP "ISTJE
       DDL  CREATE TABLE "YYYY2T"     ("KEY" CHAR(10) NOT NULL    ,"D2" CHAR(8
       DDL  CREATE INDEX "YYYY2X"     ON "YYYY2T"(KEY)     USING STOGROUP "ISTJ
****************************** END OF DB2 DATA ******************************
```

Figure 232. Show Work Statement List panel (ADB2W1S)

Use the following line commands to manage the WSL:

D   Delete the statement from the list.

I	Insert a statement into the list.
E	Edit the statement.
C	Copy this statement to the line identified by an A (after) or a B (before) line command.
M	Move this statement to the line identified by an A (after) or a B (before) line command.
A	Identifies that the destination of a move or copy operation is after this line.
B	Identifies that the destination of a move or copy operation is before this line.
R	Repeat the statement

You can issue the C and M line commands in a separate operation from the A and B line commands. If entered separately, the first line command encountered remains pending until its counterpart is encountered. While a line command is pending, any intervening line commands (such as E for edit) can be processed. However, if a line is deleted while in pending state, the operation is removed.

The following values for Type are allowed:

COM
   Comment statements

DDL
   SQL statements for data definitions, such as CREATE, ALTER, and DROP

DCL
   SQL statements for authorization changes, such as GRANT and REVOKE

DML
   SQL statements for data manipulation, such as INSERT, UPDATE and DELETE

DB2
   DB2 START, STOP, and SET commands

DSN
   DSN BIND, REBIND, and FREE commands

UTL
   DB2 utility statements

ADM
   DB2 Admin statements

Interpreting a WSL

Before running a WSL, you might want to check the contents of the WSL to see what types of statements that it contains.

About this task

Interpreting a WSL allows you to generate a report that selectively lists the different SQL statements, DB2 commands, and utility statements that the WSL contains.

To interpret a WSL:

Procedure

1. Issue the I command on the Work Statement List Library panel. The Interpret Work Statement List Options panel is displayed, as shown in the following figure.

```
 DB2 Admin ------------ Interpret Work Statement List Options ------------------

  Specify S to select Work Statement List statement types:
                                                              More:     +
    SQL:                  DB2 Utilities:          DB2 Commands:
      S DDL                 S Load/Unload            Plan/packages
        S ALTER                 LOAD                   BIND
          CREATE                UNLOAD/REORG UNLOAD     REBIND
        S DROP                S Backup/Recovery         FREE
        S COMMENT ON            COPY                  Other
        S LABEL ON              COPYTOCOPY              RUN
          SET                   MERGECOPY               START/STOP
      S DCL                     MODIFY                Other
          GRANT                 QUIESCE             Admin:
          REVOKE                REBUILD               Data set
      S DML                     RECOVER                 ALLOC
          DELETE                REPORT                  TSODELETE
          INSERT              S Other                   LISTDEF
          UPDATE                CHECK                   TEMPLATE
        Other                   DIAGNOSE                ADBSYSIN
          COMMIT                REORG               Other
          Comments              REPAIR                  ADBPAUSE
        S Other                 RUNSTATS                UTILFROM
                                STOSPACE                REXX Execs
                                Other                   Other
```

Figure 233. Interpret Work Statement List Options panel

2. Choose those statement types that you want interpreted (see the previous figure for statement types) and press Enter. The Interpret Work Statement List report is generated, as shown in the following figure. The S line command to show an object is valid only for objects that are in the catalog, such as databases, table spaces, and indexes.

```
 DB2 Admin --------- Interpret Work Statement List: WSL011 - Row 1 to 16 of 103
 Command ===>                                             Scroll ===> PAGE

 Line commands:  S - Show object  V - View statement

 Sel   Seq Action      Object Type   Qual    Name               Note
       *   *           *             *       *                  *
 --- ----- ----------  ------------- ------> -----------------> -------------
       27 SET          SQLID         ISTJEB1
       29 CREATE       DATABASE              ISTJEB1D
       31 GRANT        DATABASE              ISTJEB1D
       33 GRANT        DATABASE              ISTJEB1D
       35 GRANT        DATABASE              ISTJEB1D
       37 GRANT        DATABASE              ISTJEB1D
       46 CREATE       STOGROUP              ISTJEB1GLONG
       55 CREATE       TABLESPACE    ISTJEB1D ISTJEB1Z
       64 SET          SQLID         ISTJEB2X
       66 CREATE       TABLE         ISTJEB2X PLAN_TABLEXXXXXXXX
       68 SET          SQLID         ISTJEB1
       70 GRANT        TABLE         ISTJEB2X PLAN_TABLEXXXXXXXX
       72 GRANT        TABLE         ISTJEB2X PLAN_TABLEXXXXXXXX
       74 GRANT        TABLE         ISTJEB2X PLAN_TABLEXXXXXXXX
       76 GRANT        TABLE         ISTJEB2X PLAN_TABLEXXXXXXXX
       85 CREATE       STOGROUP              ISTJEB1G
```

Figure 234. Interpret Work Statement List report

Validating a WSL

Validating a WSL allows you to generate a report about the syntax and the impact to other objects.

About this task

Before running a WSL, you might want to have the syntax of the SQL statements checked and assess the impact that running the WSL would have on objects.

When you validate a WSL, DB2 Admin checks the syntax of each SQL statement in isolation from any other SQL statements in the WSL; it ignores any SQL statements that precede the statement currently being checked. Thus, DB2 Admin can generally report all syntactic errors but might miss semantic errors that can result from not being able to see previous statements. For example, if the name of a data type is required in a certain position in the syntax, DB2 Admin does not verify that the name of the data type is either a built-in data type or a user-defined data type that has been previously defined.

Note: For native SQL procedures, even if validation is successful, the object's existence in the body of the native SQL procedure cannot be known at procedure run time (or during procedure call).

The impact analysis portion of the validate report lists the impact to the objects by these categories:

Implicitly dropped objects
Existing objects that are implicitly dropped but not re-created by the WSL.

Explicitly dropped objects
Existing objects that are explicitly dropped but not re-created by the WSL.

Recreated objects
Existing objects that are implicitly or explicitly dropped and re-created by the WSL.

Altered objects
Existing objects that are altered by the WSL.

Created objects
Objects that did not exist and are created by the WSL.

Temporary objects
Objects that did not exist and are created and then dropped by the WSL.

Each affected object is included in only one of these categories.

After the initial validation, you can add custom validation rules for objects from Object Compare, CM Analyze, and CM Batch functions. To perform a custom validation, you must write the REXX exec that is used to perform the validation. You can specify the name of your REXX exec on the Options for Change Functions panel (ADB2PCO). The name of your REXX exec is then stored in your profile shared pool. During the validation process, the REXX exec name will be added to the JCL as the value for the STMTEXIT parameter. The REXX exit cannot connect to DB2 because the connection between DB2 Admin and DB2 is already active.

The REXX exec will be called with the following parameters:
- Statement type
- Object type

- Object qualifier
- Object name

You can decide how to perform the validation based on these parameters.

The custom validation function will continue processing until all validations are complete and all errors are reported. Validation errors are written to a separate DD file called VALOUT and use the data set name prefix that you specify. An example of a data set name is SYSADM.E344.VALOUT. Each error message from the REXX exec consists of a return code followed by a colon and the message string, as shown in the example below.

```
'08:Changes in database not allowed'
```

The return string from the REXX exec is saved to the VALOUT DD in the format shown in the example below.

```
SUPPLEMENTAL VALIDATE WORK STATEMENT LIST REPORT
================================================
Prepared on DSN7 (DB2 Release 720) by NBRON at 2006-07-08 10:48 for NBRON.WLIST.VALIDATE(SAMPLE)

ADB3036E RC=08 An error occurred while processing the ALTER DB statement: CHANGES IN DATABASE NOT ALL
```

If an object does not require validation, or if there are no errors, the REXX exec returns the code '00'.

To validate a WSL:

Procedure

1. Issue the V command on the Work Statement List Library panel. The JCL to generate the batch job to produce the Validate Work Statement List report is displayed.
2. Submit the JCL. The Validate Work Statement List report is generated and displayed, as shown in the following figure.

```
 .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
   Display  Filter  View  Print  Options  Help
 -------------------------------------------------------------------------------
 SDSF OUTPUT DISPLAY NBRONV   J0086325 DSID   105 LINE 1      COLUMNS 02- 81
 COMMAND INPUT ===>                                           SCROLL ===> PAGE
 -------------------------------------------------------------------------------
 ADB2WVL - Validate Work Statement List
 -------------------------------------------------------------------------------

          DB2 Administration Tool
          5697-L90 (C) Copyright IBM Corporation 2001, 2005.
          All rights reserved.  Licensed materials - property of IBM.
          US Government Users Restricted Rights - Use, duplication or disclosure
          restricted by GSA ADP schedule contract with IBM Corp.


 -------------------------------------------------------------------------------
 REFERENCE FOR CATALOG OBJECT STATUS
 -------------------------------------------------------------------------------
 IMPLICITLY DROPPED OBJECTS - Existing catalog objects that are implicitly
                             dropped and not recreated by the WSL.
 TEMPORARY OBJECTS          - Objects that are created and dropped during
                             execution of the WSL.  Temporary objects do not
                             exist in the catalog before or after WSL execution.
 CREATED OBJECTS            - Objects that are created by the WSL that did not
                             exist in the catalog.
 EXPLICITLY DROPPED OBJECTS - Existing catalog objects that are explicitly
                             dropped and not recreated by the WSL.
 ALTERED OBJECTS            - Existing catalog objects that are modified by
                             ALTER statements in the WSL.
 RECREATED OBJECTS          - Existing catalog objects that are implicitly or
                             explicilty dropped and later recreated by the WSL.
 -------------------------------------------------------------------------------


 VALIDATE WORK STATEMENT LIST REPORT
 ===================================

 Prepared on DSN7 (DB2 Release 720) by NBRON at 2006-07-08 10:48
 for NBRON.WLIST.VALIDATE(SAMPLE)

 SQL error in PREPARE for statement:
  CREATE SEQUENCE ORDER_SEQ               START WITH 1             INCREMEN
  DSNT408I SQLCODE = -104, ERROR:  ILLEGAL SYMBOL "START". SOME SYMBOLS THAT
          MIGHT BE LEGAL ARE: FOR
  DSNT418I SQLSTATE   = 42601 SQLSTATE RETURN CODE
  DSNT415I SQLERRP    = DSNHPARS SQL PROCEDURE DETECTING ERROR
  DSNT416I SQLERRD    = 0  0  0  -1  40  0 SQL DIAGNOSTIC INFORMATION
  DSNT416I SQLERRD    = X'00000000'  X'00000000'  X'00000000'  X'FFFFFFFF'
          X'00000028'  X'00000000' SQL DIAGNOSTIC INFORMATION

 Error processing Database  ABCDE in a ALTER statement:Object does not exist
 Error processing Table DSN8720.ABCDTB in a ALTER statement:Object does not exist
 Error processing Table DSN8720.DEPT in a ALTER statement:Object does not exist
 Error processing Table DSN8720.ABCDTB in a ALTER statement:Object does not exist
 Error processing Index DSN8720.ABCDIX in a ALTER statement:Object does not exist
 Error processing Index DSN8720.XDEPT1 in a ALTER statement:Object does not exist
 Error processing Sequence NBRON.org_seq in a ALTER statement:Object does not exist
 Error processing Sequence VNDSHL2.SEQ14 in a CREATE statement:Object already exists
 Error processing Sequence VNDSHL2.SEQ13 in a DROP statement:Object does not exist
 .
 .
 .
```

Figure 235. Validate Work Statement List report (1 of 2)

```
.
.
.
IMPLICITLY DROPPED OBJECTS
--------------------------
  Referential constraint AHXTOOLS.PROJACT
  Referential constraint AHXTOOLS.PROJ
  Referential constraint AHXTOOLS.PROJ
  Referential constraint AHXTOOLS.DEPT
  Referential constraint AHXTOOLS.PROJ
  Referential constraint AHXTOOLS.PROJACT
  Referential constraint AHXTOOLS.DEPT

ALTERED OBJECTS
---------------
  Function NBRON.SPECIFICFFFF1


TEMPORARY OBJECTS
-----------------
  Sequence NBRON.org_seq
  Table Space DSN8D72A.DSN8S72D
  Table DSN8720.DEPT
  Table DSN8720.ABCDTB


CREATED OBJECTS
---------------
  Table NBRON.TBDSN80


RECREATED OBJECTS
-----------------
  Table QUADPB02.TBADPB02
  Table Space DBADPB02.TPADPB01
  View QUADPB02.VWADPB02
  View QUADPB02.VWADPB04
  View QUADPB02.VWADPB05
  View QUADPB02.VWADPB06
  View QUADPB02.VWADPB09
  View QUADPB02.VWADPB12
  View QUADPB02.VWADPB14
  View QUADPB02.VWADPB15
  View QUADPB02.VWADPB16
  View QUADPB02.VWADPB17
  View QUADPB02.VWADPB18
  View QUADPB02.VWADPB19
  Index QUADPB02.IPADPB01
  Index QUADPB02.IPADPB02
  Referential constraint QUADPB02.TBADPB02 QUADPB02.TBADPB01 FKADPB03
  Referential constraint QUADPB02.TBADPB01 QUADPB02.TBADPB02 FKADPB02
  Referential constraint QUADPB02.TBADPB04 QUADPB02.TBADPB02 FKADPB04
  Referential constraint QUADPB02.TBADPB05 QUADPB02.TBADPB02 FKADPB07
```

Figure 236. Validate Work Statement List report (2 of 2)

What to do next

After completing the initial validation, you can impose additional custom
validation rules to DB2 objects.

- For Object Compare and CM Analyze, use the STMTEXIT parameter. The value
  for this parameter can be found on the Options for Change Functions panel
  (ADB2PCO). You can manually update that value by changing it in the JCL, then
  submitting the JCL.

```
//ADBWL    DD DISP=SHR,
//            DSN=SYSADM.WSL
//CAT      DD DSN=&&CATOUT2,
//            DCB=(LRECL=16800,RECFM=VB,DSORG=PS),
//            SPACE=(TRK,(15,15),RLSE),
```

```
|              //          UNIT=SYSDA,
|              //          DISP=(NEW,PASS)
|              //ADBUEXEC DD DISP=SHR,DSN=SPF.PRODUCT.ISPTLIB
|              //ADBUEXE1 DD DISP=SHR,DSN=SPF.PRODUCT.ISPTLIB
|              //VALOUT   DD SYSOUT=*
|              //IN       DD *
|                DB2SYS   = DSNB,
|                DB2ALOC  =
|                DB2SERV  = DSNB
|                DB2AUTH  = 'S22957'
|                DB2REL   = 1115
|                ADBTEST  = YES,
|                PLAN     = ADB,
|                ADBASUSR = ,
|                ADBASUSB = NO,
|                SRCWSLST = TEST3,
|                SRCWSLIB = SYSADM.WSL,
|                STMTEXIT = TEST
|              /*
|              /*
```
| • For CM Batch, use the VALIDATION_STMTEXIT parameter. There is no default
| value for this parameter.

```
|              //* DB2 ADMIN ISPF BATCH
|              //*
|              //T03REG EXEC GOCCM,SSID=DSNB,PLAN=ADBDEV,GRP=UB2DEV0,USRGRP=S45801
|              //GOCCM.PARMS  DD *
|               ACTION_IMPORT_CHANGE='N'
|               ACTION_ANALYZE_CHANGE='Y'
|               CHANGE_OWNER='S22957'
|               CHANGE_NAME='IMPORT_DSTCHANGE'
|               PDS_FOR_WSL='SYSADM.WSL'
|               VALIDATE_STMTEXIT='TEST'
|               CHANGE_COMMENT=''
|               ;
```

Running a WSL

You can run a WSL.

About this task

To run a WSL:

Procedure

1. Issue the R (Run in batch) command or the O (Run online) command on the
   Work Statement List Library panel for the WSL that you want to run. If you
   choose to run in batch, the JCL to generate the batch job is displayed.
2. Submit the JCL.
3. If the WSL included a LOAD operation, review the Load Summary Report in
   LOADRPT, which indicates whether records were discarded when data was
   loaded. When a Load Summary Report step exists, SYSPRINT output from the
   preceding ADBTEP2 step is recorded in ADBPRINT of the Load Summary
   Report step. If the WSL does not include a LOAD, ADBTEP2 messages are
   recorded in SYSPRINT of the ADBTEP2 step.

Load summary report

Checking the load summary report (located in LOADPRT) at the end of a WSL run
is easier than scanning the WSL execution log and checking for instances of
load-generated discard records.

The load summary report helps you ensure that no data was unexpectedly lost.

The load summary report contains the following information:
- The name of the object
- The number of input records
- The number of records that were loaded
- The number of records that were discarded

The example in the following figure shows a load summary report in which the number of input and loaded records for three tables were the same, but records were discarded for another table.

```
15697-L90 IBM DB2 Administration Tool for Z/OS        Load Summary Report for Worklist(ST8)


Table owner        Table name           Input      Loaded   Discarded Status
------------------ ------------------ ----------- ----------- ----------- --------
"SYSADM"           "TBADAS01"              1255        1255           0 ********
"SYSADM"           "TBADAS02"               855         799          56 discards
"SYSADM"           "TBADAS03"              2033        2033           0 ********
"SYSADM"           "TBADAS04"              1444        1444           0 ********
```

Figure 237. Example of load summary report

When the report contains a large number of rows, you will need to scroll through the report to see all of the information in the report. When the table name exceeds the number of characters that can be displayed in the **Table Name** field, a footnote suffix is added to the table name, and the full table name is displayed at the bottom of the report. The following example shows the format that is used to display long table names:

```
15697-L90 IBM DB2 Administration Tool for Z/OS        Load Summary Report for Worklist(ST9)


Table owner        Table name           Input      Loaded   Discarded Status
------------------ ------------------ ----------- ----------- ----------- --------
"SYSADM"           "TBADAS0190123(*1)      1006        1006           0 ********
"SYSADM"           "TBADAS0290123(*2)        75          75           0 ********
"SYSADM"           "TBADAS0390123(*3)      4031        4031           0 ********
"SYSADM"           "TBADAS0490123(*4)      2444        2444           0 ********

Footnotes:

(*1)

 "TBADAS019012345678901234567890"

(*2)

 "TBADAS029012345678901234567890"

(*3)

 "TBADAS039012345678901234567890"

(*4)

 "TBADAS049012345678901234567890"
```

Figure 238. Example of load summary report with long table names

Restarting a WSL

If your WSL stops running due to an error, you can restart it.

Before you begin

Ensure that any errors in the WSL have been corrected.

About this task

If a WSL fails in the middle of a run, you can run it again. When you restart the WSL, the Specify Restart Information panel is displayed, as shown in the following figure.

```
DB2 Admin ----------- Specify Restart Information: BASEPRCB -- Row 1 to 2 of 2
Command ===>                                              Scroll ===> PAGE

Commands: NEXT
Line commands:
B - Checkpoint    V - Edit Restart Info    R - Toggle Restart Report Only
C - Toggle Ckpt Env    I - Toggle Input Env

                      Ckpt --- ENV --- Report  User
Sel    Suffix    Restart Fnd   Ckpt  Input  Only   Restart
       *         *       *     *     *      *      *
----- -------- ------- ----- ----- ------ ------ --------
                Y       Y     Y     Y      N      BOB
```

Figure 239. Specify Restart Information panel

On the Specify Restart Information panel, you can restart the WSL. There are two types of restarts:

- System-controlled
- User-controlled

A system-controlled restart is automated by DB2 Admin, and restarts the WSL from the point where it failed.

A user-controlled restart allows you to restart the WSL from a point different than where it failed.

Procedure

Choose one of the following restart options:

Option	Description
System-controlled restart (default)	To restart the WSL from the point of failure: 1. Enter Y in the **Restart** column of the Specify Restart Information panel. 2. Issue the CONTINUE command.

Option	Description

User-controlled restart	To restart the WSL from a point that you specify:
	
	1. In your WSL, add the line `--#RESTART <string>` at the point that you want your WSL to restart from. The string identifier can be anything except YES, NO, FORCE, or a pure numeric value.
	Note: You can add as many user-defined restart points to a WSL as you want, but only one will be used for restart.
	
	2. Issue the V line command on the Specify Restart Information panel.
	
	3. In the **Restart** column, enter U.
	
	4. In the **User Restart** column, enter the string identifier that you added to your WSL in step 1 indicating the point of restart.
	
	5. Issue the CONTINUE command.
	Note: Alternatively, you can specify the restart point directly in the batch statement list. To restart from the batch statement list, add the line `--#RESTART <string>` at the point that you want the statement to restart from. In addition, you must add RESTART AT `<string>` with a terminator as the first line, like the following example:
	<pre>//SYSIN DD *
	RESTART AT BOB;
	DROP TABLE
	    POSTO.PPP1;
	COMMIT;
	--#RESTART BOB
	DROP TABLE POSTO.PPP2;
	COMMIT;</pre>

Restart report only	To simulate a restart without actually running a restart, so that you can see the results before deciding whether to run a restart: 1. Issue the R line command next to the WSL that you want to restart. 2. Issue the CONTINUE command.

Restarting a WSL that was run by another user

You can restart a WSL that was run by another user but did not complete successfully.

About this task

To restart a WSL that was run by another user:

Procedure

1. Determine the user ID of the user who ran the WSL. You can find the user ID in the checkpoint table.

2. Issue the R (Run in batch) command on Work Statement List Library panel for the WSL that you want to restart. The JCL to generate the batch job is displayed.

3. Edit the batch job at the ADBTEP2 restart job step and specify the USER parameter with the user ID of the user who originally ran the WSL. For example, if a user with user ID SYSADM ran the WSL, the following portion of code shows how the edited JCL would look with the USER parameter added:

```
000036  RUN  PROGRAM(ADBTEP2) PLAN(ADBTEP2) -
000037    LIB('ADB.QA260.ISPLLIB')  -
000038    PARMS('/WORKLIST(JTKZ) SSID(V81A)  -
 ''''''     USER(SYSADM) -
000039  RESTART(YES),BINDERROR(MAXE)')
```

Important: The USER and CHANGEID parameters are mutually exclusive. Ensure that the JCL does not include a CHANGEID parameter.

4. Submit the JCL.

Sample scenario for creating and using a work statement list

This scenario shows how to use DB2 Object Comparison Tool to create a WSL.

In this scenario, two databases are used, each with two tables. DB2 Object Comparison Tool produces the tasks that upgrade the older, outdated database to the new database. This sample directs these tasks to a WSL. The following figure shows the DB2 Object Comparison Tool after defining the inputs and the masking that is required, and proceeding to the Step 5. Generate Compare Jobs panel.

```
Compare -------------------- Generate Compare Jobs --------------------------
Option ===>

 Specify the following for DB2 Object Comparison Tool:
                                                         More:     +
 Worklist information:
   Worklist name . . . . . . : ROYCDOC1  (also used as middle qualifier in DSNs)

 Compare options:
   Suppress DROP of objects  : N        (Yes/No)
   Suppress DROP of columns  : N        (Yes/No)
   Suppress adding columns . : N        (Yes/No)

 Change reporting options  . : N        (Yes/No)

 Data set information:
   PDS for batch jobs  . . . : DOCM.CNTL
   Prefix for data sets  . . : ROYC
   Unit type permanent ds  . : SYSDA
   Unit type unload ds . . . : SYSDA    Serial (tape) device : N (Y/N)

 Options:
   Single compare job  . . . : Y        (Yes/No)
   Member name of single job : COMPARE  (default COMPARE )
   Generate apply jobs . . . : N        (Yes/No)
   As work statement list  . : Y        (Yes/No to append to work stmt list)

 Optional jobs after reload:
   Run CHECK DATA  . . . . . : Y        (Yes/No)
   Run RUNSTATS  . . . . . . : Y        (Yes/No)
   Take an image copy  . . . : Y        (Yes/No)

 BP - Change batch job parameters
```

Figure 240. DB2 Object Comparison Tool — Generate Compare Jobs panel

The new WSL name is ROYCDOC1 and the **As work statement list** field indicates
that the job should be saved as a WSL. Next, a panel prompts for the data set in
which to store the new WSL. If the data set does not exist, it is created. A DB2
Object Comparison Tool JCL job is now generated for this new WSL. Running this
job produces the WSL that can be used to upgrade the old tables to the new tables.

The following figure shows the result of selecting option 1 on panel ADB2W
(option W from the Main Menu) to show the list of WSLs, including the new WSL
just created.

```
DB2 Admin ---- Work Statement List Library: ROYC.WORKLIST -- Row 1     of 1
Command ===>                                               Scroll ===> PAGE

Line commands:
Line commands:
S - Show  R - Run in batch  D - Delete  C - Copy  A - Append  Q - Clone
I - Interpret  V - Validate  E - Edit

  Name       Prompt        Size   Created         Changed         ID
- ------------------------------------------------------------------------------
  ROYCDOC1
  **End**
```

Figure 241. Work Statement List Library panel (ADB2W1)

Figure 242 on page 373 and Figure 243 on page 374 show the contents of the new
WSL, using the SHOW line command.

The TYPE column specifies the statement type (DDL statement, DB2 command,
DB2 utility, etc.) for statements that are placed in the batch statement list when

running the WSL. The ADM type statements are control statements that can control the number of jobs created when the WSL is run.

```
DB2 Admin ----------- Show Work Statement List: ROYCDOC1 --- Row 1 to 14 of 83
Command ===>                                              Scroll ===> PAGE

Line commands:
 D - Delete  I - Insert  E - Edit  C - Copy  M - Move  A - After  B - Before
 R - Repeat

Select Type Statement
       *    *
------ ---- ----------------------------------------------------------------------
       COM  -- Created by ROYC on 2002/07/16 at 16:49
       COM  Generated by Compare Apply by ROYC on 2002/07/16 at 16:49
       ADM  PARALLEL UNLOAD
       ADM  JOB
       DB2  -STA DB(POST) SPACE(POSTTS1) ACCESS(RO)
       UTL  TEMPLATE UTLPUNCH DSN 'ROYC.ROYCDOC1.CNTL.PPP1'..        UNIT SYSD
       UTL  TEMPLATE SYSREC DSN 'ROYC.ROYCDOC1.UNLD.PPP1'..          UNIT SYSDA
       UTL  UNLOAD DATA" FROM TABLE "POSTO"."PPP1" PUNCHDDN(UTLPUNCH)
       DML  TSODELETE 'ROYC.ROYCDOC1.CNTLC.PPP1';..TSODELETE 'ROYC.ROYCDOC1.UNL
       TSO  ALLOC DD(DDLIN) DUMMY
       TSO  ALLOC DD(DDLOUT) DUMMY
       TSO  ALLOC DD(CNTLI001)..     DS('ROYC.ROYCDOC1.CNTL.PPP1') SHR
       TSO  ALLOC DD(CNTLO001)..     DS('ROYC.ROYCDOC1.CNTLC.PPP1')..    LIK
       TSO  ALLOC DD(DATAI001)..     DS('ROYC.ROYCDOC1.UNLD.PPP1') SHR
       TSO  ALLOC DD(DATAO001)..     DS('ROYC.ROYCDOC1.UNLDC.PPP1') USING(DATA
       ADM  ADMIN ALTER CONVERT POSTO.PPP1
       ADM  ENDJOB
       ADM  JOB
       DB2  -STA DB(POST) SPACE(POSTTS2) ACCESS(RO)
       UTL  TEMPLATE UTLPUNCH DSN 'ROYC.ROYCDOC1.CNTL.PPP2'..        UNIT SYSD
       UTL  TEMPLATE SYSREC DSN 'ROYC.ROYCDOC1.UNLD.PPP2'..          UNIT SYSDA
       UTL  UNLOAD DATA FROM TABLE "POSTO"."PPP2" PUNCHDDN(UTLPUNCH)
       DML  TSODELETE 'ROYC.ROYCDOC1.CNTLC.PPP2';..TSODELETE 'ROYC.ROYCDOC1.UNL
       TSO  ALLOC DD(DDLIN) DUMMY
       TSO  ALLOC DD(DDLOUT) DUMMY
       TSO  ALLOC DD(CNTLI001)..     DS('ROYC.ROYCDOC1.CNTL.PPP2') SHR
       TSO  ALLOC DD(CNTLO001)..     DS('ROYC.ROYCDOC1.CNTLC.PPP2')..    LIK
       TSO  ALLOC DD(DATAI001)..     DS('ROYC.ROYCDOC1.UNLD.PPP2') SHR
       TSO  ALLOC DD(DATAO001)..     DS('ROYC.ROYCDOC1.UNLDC.PPP2') USING(DATA
       ADM  ADMIN ALTER CONVERT POSTO.PPP2
       ADM  ENDJOB
       ADM  ENDPARALLEL
       DDL   DROP TABLE POSTO.PPP1
       DML   COMMIT
       DDL   DROP TABLE POSTO.PPP2
       DML   COMMIT
       DB2  -STA DB(POST) SPACE(POSTTS1)
       DB2  -STA DB(POST) SPACE(POSTTS2)
       DDL   CREATE TABLE POSTO.PPP1..     (EMP                CHAR(6) FOR S
       DML   COMMIT
       DDL   CREATE TABLE POSTO.PPP2..     (EMP                CHAR(6) FOR S
       DML   COMMIT
       DDL   CREATE INDEX POSTO.PPP1X..    ON POSTO.PPP1..     (EMP
       DML   COMMIT
       DDL   CREATE INDEX POSTO.PPP2X..    ON POSTO.PPP2..     (EMP
       DML   COMMIT
       ADM  PARALLEL RELOAD
       ADM  JOB
...
```

Figure 242. The contents of the new WSL (part 1)

```
...
      UTL   TEMPLATE SYSREC DSN 'ROYC.ROYCDOC1.UNLDC.PPP1' DISP(SHR)
      UTL   TEMPLATE UTLDISC DSN 'ROYC.ROYCDOC1.SDISC.PPP1'..          UNIT SYSD
      UTL   TEMPLATE UTLERR DSN 'ROYC.ROYCDOC1.SERR.PPP1'..          UNIT SYSDA.
      UTL   TEMPLATE UTLMAP DSN 'ROYC.ROYCDOC1.SMAP.PPP1'..          UNIT SYSDA
      UTL   TEMPLATE UTLUT1 DSN 'ROYC.ROYCDOC1.SUT1.PPP1'..          UNIT SYSDA
      UTL   TEMPLATE UTLOUT DSN 'ROYC.ROYCDOC1.SOUT.PPP1'..          UNIT SYSDA
      UTL   UTILFROM ROYC.ROYCDOC1.CNTLC.PPP1..          ADD(SORTNUM 8 SORTDEVT
      UTL   TEMPLATE UTLUT1 DSN 'ROYC.ROYCDOC1.SUT1.PPP1'..          UNIT SYSDA
      UTL   TEMPLATE UTLOUT DSN 'ROYC.ROYCDOC1.SOUT.PPP1'..          UNIT SYSDA
      UTL   TEMPLATE UTLERR DSN 'ROYC.ROYCDOC1.SERR.PPP1'..          UNIT SYSDA
      UTL   CHECK DATA TABLESPACE POST.POSTTS1..  ERRDDN(UTLERR) WORKDDN(UTLUT1
      UTL   RUNSTATS TABLESPACE POST.POSTTS1..  TABLE("POSTO"."PPP1")..  INDEX(
      UTL   TEMPLATE SYSCOPY DSN 'ROYC.DSN7.IC.POST.POSTTS1(+1)'..          UNIT
      UTL   COPY TABLESPACE POST.POSTTS1 COPYDDN(SYSCOPY)
      UTL   MODIFY RECOVERY TABLESPACE POST.POSTTS1 DSNUM ALL..  DELETE AGE(35)
      ADM   ENDJOB
      ADM   JOB
      UTL   TEMPLATE SYSREC DSN 'ROYC.ROYCDOC1.UNLDC.PPP2' DISP(SHR)
      UTL   TEMPLATE UTLDISC DSN 'ROYC.ROYCDOC1.SDISC.PPP2'..          UNIT SYSD
      UTL   TEMPLATE UTLERR DSN 'ROYC.ROYCDOC1.SERR.PPP2'..          UNIT SYSDA.
      UTL   TEMPLATE UTLMAP DSN 'ROYC.ROYCDOC1.SMAP.PPP2'..          UNIT SYSDA
      UTL   TEMPLATE UTLUT1 DSN 'ROYC.ROYCDOC1.SUT1.PPP2'..          UNIT SYSDA
      UTL   TEMPLATE UTLOUT DSN 'ROYC.ROYCDOC1.SOUT.PPP2'..          UNIT SYSDA
      UTL   UTILFROM ROYC.ROYCDOC1.CNTLC.PPP2..          ADD(SORTNUM 8 SORTDEVT
      UTL   TEMPLATE UTLUT1 DSN 'ROYC.ROYCDOC1.SUT1.PPP2'..          UNIT SYSDA
      UTL   TEMPLATE UTLOUT DSN 'ROYC.ROYCDOC1.SOUT.PPP2'..          UNIT SYSDA
      UTL   TEMPLATE UTLERR DSN 'ROYC.ROYCDOC1.SERR.PPP2'..          UNIT SYSDA
      UTL   CHECK DATA TABLESPACE POST.POSTTS2..  ERRDDN(UTLERR) WORKDDN(UTLUT1
      UTL   RUNSTATS TABLESPACE POST.POSTTS2..  TABLE("POSTO"."PPP2")..  INDEX(
      UTL   TEMPLATE SYSCOPY DSN 'ROYC.DSN7.IC.POST.POSTTS2(+1)'..          UNIT
      UTL   COPY TABLESPACE POST.POSTTS2 COPYDDN(SYSCOPY)
      UTL   MODIFY RECOVERY TABLESPACE POST.POSTTS2 DSNUM ALL..  DELETE AGE(35)
      ADM   ENDJOB
      ADM   ENDPARALLEL
      COM   End of Compare Apply statements
****************************** END OF DB2 DATA ******************************
```

Figure 243. The contents of the new WSL (part 2)

When you run the WSL (by issuing the R line command on the option 1 panel), you are prompted for a library name, a prefix to use for the job name, and whether the job name should equal the member name. The following figure shows the jobs that are created when you select Run Work Statement List.

```
  Menu  Functions  Utilities  Help
 SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS
 EDIT     ROYC.DOCM.CNTL                                 Row 00001 of 00006
 Command ===>                                            Scroll ===> CSR
   Name        Prompt        Size   Created         Changed          ID
 . COMPARE                   188    2002/07/16  2002/07/16 17:55:30  ROYC
 . ROYCR001                   68    2002/07/16  2002/07/16 18:33:06  ROYC
 . ROYCR002                   68    2002/07/16  2002/07/16 18:33:07  ROYC
 . ROYCU001                   64    2002/07/16  2002/07/16 18:33:04  ROYC
 . ROYCU002                   62    2002/07/16  2002/07/16 18:33:04  ROYC
 . ROYC2                      82    2002/07/16  2002/07/16 18:33:05  ROYC
   **End**
```

Figure 244. The jobs that are generated from running the WSL.

The Run command produced five jobs. This example is changing two tables; therefore, two unload jobs (ROYCU001 and ROYCU002) are created. These two jobs can be run in parallel. The ROYC2 job performs all the DDL tasks and can be run after the unload jobs have successfully completed. The final two jobs, ROYCR001 and ROYCR002, reload the data and can be run in parallel. The COMPARE job is shown in the previous figure but does not need to be in the same library as the other WSL jobs.

Figure 245 on page 376 and Figure 246 on page 377 show the ROYC2 job in detail. The following statements in this job are important to understand:

- RUN PROGRAM(ADBTEP2) PLAN(ADBTEP2) specifies that the DB2 Admin Batch Restart Program (ADBTEP2) is to be run.
- The library that contains ADBTEP2 is specified in the line: LIB('ADBB10.SADBLLIB'). This library cannot be in the STEPLIB because the STEPLIB must be APF authorized to run DB2 utilities.
- The WSL name appears as the first part of the WORKLIST parameter in the line PARM('/WORKLIST(ROYCDOC1.2),SSID(DSN7)'). Also, the SSID parameter is mandatory if DB2 commands or utilities are being executed.
- The input to ADBTEP2 is provided by the SYSIN DD name, which is referred to as a *batch statement list*. This contains the executable statements derived from the WSL.

```
****** ***************************** Top of Data ******************************

DB2 Admin: Edit generated JCL

//ROYCDOC1   JOB (ROYC,B240,090,D783),&SYSUID,
//*        RESTART=STEPNAME, <== FOR RESTART REMOVE * AND ENTER STEP NAME
//      MSGCLASS=H,TIME=(2),MSGLEVEL=(1,1),NOTIFY=&SYSUID,
//      USER=&SYSUID,REGION=8M
//*
//          CLASS=U
//*
/*JOBPARM S=SY4A
//*
//*
//***********************************************************************
//*
//* DB2 BATCH MONITOR
//*
//* DB2 ADMIN GENERATED BATCH JOB.
//*
//**********************************************************ADB2WL4**
//DB2B EXEC PGM=IKJEFT01,DYNAMNBR=100
//STEPLIB  DD DISP=SHR,DSN=DSN.DSN7.SDSNEXIT
//         DD DISP=SHR,DSN=DSN.DSN7.SDSNLOAD
//SYSEXEC  DD DISP=SHR,DSN=ADB4DEVT.EXEC
//         DD DISP=SHR,DSN=GOC2BASE.EXEC
//         DD DISP=SHR,DSN=ADBB10.SADBEXEC
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//UTPRINT  DD SYSOUT=*
//SYSTSIN  DD *
DSN SYSTEM(DSN7)
  RUN PROGRAM(ADBTEP2) PLAN(ADBTEP2) -
  LIB('ADBB10.SADBLLIB') -
  PARM('/WORKLIST(ROYCDOC1.2),SSID(DSN7)')
END
//SYSIN    DD *
  DROP TABLE POSTO.PPP1;
  COMMIT;
  DROP TABLE POSTO.PPP2;
  COMMIT;
-STA DB(POST) SPACE(POSTTS1);
-STA DB(POST) SPACE(POSTTS2);
  CREATE TABLE POSTO.PPP1
    (EMP                CHAR(6) FOR SBCS DATA WITH DEFAULT NULL ,
     PROJ               CHAR(3) FOR SBCS DATA WITH DEFAULT NULL )
   IN POST.POSTTS1
   AUDIT NONE
   DATA CAPTURE NONE
   CCSID       EBCDIC;
  COMMIT;
  CREATE TABLE POSTO.PPP2
    (EMP                CHAR(6) FOR SBCS DATA WITH DEFAULT NULL ,
     DEPT               CHAR(3) FOR SBCS DATA WITH DEFAULT NULL )
   IN POST.POSTTS2
   AUDIT NONE
   DATA CAPTURE NONE
   CCSID       EBCDIC;
  COMMIT;

...
```

Figure 245. The resulting job: ROYC2 (part 1)

```
...

  CREATE INDEX POSTO.PPP1X
    ON POSTO.PPP1
     (EMP                  ASC)
    USING STOGROUP SYSDEFLT
    PRIQTY 12 SECQTY 12
    ERASE  NO
    FREEPAGE 0 PCTFREE 10
    GBPCACHE CHANGED
    BUFFERPOOL BP1
    CLOSE YES
    COPY NO
    PIECESIZE 2 G;
  COMMIT;
  CREATE INDEX POSTO.PPP2X
    ON POSTO.PPP2
     (EMP                  ASC)
    USING STOGROUP SYSDEFLT
    PRIQTY 12 SECQTY 12
    ERASE  NO
    FREEPAGE 0 PCTFREE 10
    GBPCACHE CHANGED
    BUFFERPOOL BP1
    CLOSE YES
    COPY NO
    PIECESIZE 2 G;
  COMMIT;
 /*
```

Figure 246. The resulting job: ROYC2 (part 2)

Running WSL with the utility template for LOBs

You can run work statement lists (WSLs) with LOBs by using the utility template
for LOBs, or by using a customization skeleton, or you can run WSLs by default.

If you use the utility template for LOBs, the Run WSL function (like other
functions such as ALT and MIG) will add an ADM statement (ADMIN
LOBTEMPLATE) to indicate the existence of a LOB column or columns in the table
or tablespace that is involved in the next UNLOAD statement.

The LOBTEMPLATE statement format is

ADMIN LOBTEMPLATE <n> DSN <DSNPrefix>...<and other attributes like UNIT, SPACE...>

where

<n>　　Indicates the existence of *n* number of LOB columns in the next unload.

<DSNPrefix>
　　　　The dataset prefix, which can have a maximum length of 35 bytes.

When the Run WSL function reads each ADMIN LOBTEMPLATE statement, the
Run WSL function performs the following steps:
1. Generates a unique name for the template.

   For example, the following name: ADBL<nnnn>

   where

   ADB　　Indicates that it is an admin template.

L   Indicates that it is a LOB template.

nnnn   Is a running sequence number for each LOB template.

2. Multiplies the given template statement into *n* templates by adding a name for the template and adding a suffix for the data set, as shown in the following example:

```
ADMIN LOBTEMPLATE  <n> DSN <DSNPrefix>...<and other attributes like UNIT, SPACE...>
```

The Run WSL function then replaces the preceding statement with the following set of statements:

```
ADMIN LOBTEMPLATE  ADBL1 DSN <DSNPrefix>...<and other attributes like UNIT, SPACE...>
ADMIN LOBTEMPLATE  ADBL2 DSN <DSNPrefix>...<and other attributes like UNIT, SPACE...>
.
.
.
TSODELETE ''SMITHS..&SSID..&DB..&SN..ADBLn'
ADMIN LOBTEMPLATE  ADBLn DSN <and other attributes like UNIT, SPACE...>
```

The Run WSL function places the templates before the corresponding UNLOAD statement by replacing the ADMIN LOBTEMPLATE statement that was generated by the DB2 Admin functions.

The ADMIN LOBTEMPLATE statement triggers ADBTEP2 to make the necessary modifications to the UNLOAD statement.

```
ADB2W1S n ----------- Show Work Statement List: LOBDB ------ Row 3 to 35 of 81
Command ===>                                               Scroll ===> CSR

Line commands:
 D - Delete  I - Insert  E - Edit  C - Copy  M - Move  A - After  B - Before
 R - Repeat

Select Type Statement
       *    *
------ ---- -------------------------------------------------------------------->
       ADM  PARALLEL UNLOAD
       ADM  JOB
       DB2  -STA DB(LOBDB) SPACE(KAVTS) ACCESS(RO)
       TSO  TSODELETE 'SMITHS.DB8A.LOBDB.CNT.T0001'
       TSO  TSODELETE 'SMITHS.DB8A.LOBDB.ULD.T0001'
       ADM  ADMIN LOBTEMPLATE 2 DSN 'SMITHS.&SSID..&DB..&SN' UNIT(SYSDA)
       UTL  TEMPLATE UTLPUNCH DSN 'SMITHS.DB8A.LOBDB.CNT.T0001'..        UNIT
       UTL  TEMPLATE UTLREC DSN 'SMITHS.DB8A.LOBDB.ULD.T0001'..          UNIT S
       UTL  UNLOAD DATA FROM TABLE.."SMITHS"."LOB2TB"..UNLDDN UTLREC..PUNCHDDN(
       <...more statements...>
       COM  -- End of Compare Apply statements
***************************** END OF DB2 DATA *****************************
```

Figure 247. Show Work Statement List: LOBDB (ADB2W1S)

Running WSL with the utility template for unloading XML data

You can run work statement lists (WSLs) with XML by using the utility template for XML, or by using a customization skeleton, or you can run WSLs by default.

If you use the utility template for XML, the Run WSL function will repeat the ADMIN XMLTEMPLATE *n* statement *n* times.

The XMLTEMPLATE statement format is

```
ADMIN XMLTEMPLATE  <n> DSN <DSNPrefix>...<and other attributes like UNIT, SPACE...>
```

where

<n>   Indicates the existence of *n* number of XML columns in the next unload.

<DSNPrefix>
>The dataset prefix, which can have a maximum length of 35 bytes.

When the Run WSL function reads each ADMIN XMLTEMPLATE statement, the Run WSL function performs the following steps:

1. Appends a qualifier as needed for the template. Ensure that your data set is unique after the qualifier is appended.

   For example, the following name: ADBX<nnnn>

   where

   ADB   Indicates that it is an admin template.

   X   Indicates that it is an XML template.

   nnnn   Is a running sequence number for each XML template.

2. Repeats the given template statement into *n* templates by adding a name for the template and adding a suffix for the data set, as shown in the following example:

   ```
   ADMIN XMLTEMPLATE  <n> DSN <DSNPrefix>...<and other attributes like UNIT, SPACE...>
   ```

   The Run WSL function then replaces the preceding statement with the following set of statements:

   ```
   ADMIN XMLTEMPLATE  ADBX1 DSN <DSNPrefix>...<and other attributes like UNIT, SPACE...>
   ADMIN XMLTEMPLATE  ADBX2 DSN <DSNPrefix>...<and other attributes like UNIT, SPACE...>
   .
   .
   .

   ADMIN XMLTEMPLATE  ADBXn DSN <and other attributes like UNIT, SPACE...>
   ```

   The Run WSL function places the templates before the corresponding UNLOAD statement by replacing the ADMIN XMLTEMPLATE statement that was generated by the DB2 Admin functions.

   The ADMIN XMLTEMPLATE statement triggers ADBTEP2 to make the necessary modifications to the UNLOAD statement.

   Attention: The data set name pattern will be modified to include an additional qualifier when multiple XML or LOB columns exist in the object being unloaded and &TS or &SN are not included and the unload method chosen is DB2. If the unload method chosen is HPU, this check or modification is not performed as HPU will detect a data set collision and fail the unload.

   Restriction: If ADBTEP2 encounters too few XML templates for the object being unloaded, it will issue message ADB5224E and end processing.

```
ADB2W1S n ----------- Show Work Statement List: LOBDB ------ Row 3 to 35 of 81
Command ===>                                               Scroll ===> CSR

Line commands:
 D - Delete  I - Insert  E - Edit  C - Copy  M - Move  A - After  B - Before
 R - Repeat

Select Type Statement
       *     *
------ ---- -------------------------------------------------------------------->
       ADM  PARALLEL UNLOAD
       ADM  JOB
       DB2  -STA DB(LOBDB) SPACE(KAVTS) ACCESS(RO)
       TSO  TSODELETE 'SMITHS.DB8A.LOBDB.CNT.T0001'
       TSO  TSODELETE 'SMITHS.DB8A.LOBDB.ULD.T0001'
       ADM  ADMIN XMLTEMPLATE 2 DSN 'SMITHS.&SSID..&DB..&SN' UNIT(SYSDA)
       UTL  TEMPLATE UTLPUNCH DSN 'SMITHS.DB8A.LOBDB.CNT.T0001'..        UNIT
       UTL  TEMPLATE UTLREC DSN 'SMITHS.DB8A.LOBDB.ULD.T0001'..          UNIT S
       UTL  UNLOAD DATA FROM TABLE.."SMITHS"."LOB2TB"..UNLDDN UTLREC..PUNCHDDN(
       <...more statements...>
       COM  -- End of Compare Apply statements
***************************** END OF DB2 DATA ********************************
```

Figure 248. Show Work Statement List: XMLDB (ADB2W1S)

Using DB2 High Performance Unload within a work statement list

When using the DB2 Admin Alter ALT and Migrate functions, you can use DB2 High Performance Unload (HPU) within a work statement list.

In addition, when using ALTER table space redefine against a single table space, you can use HPU as the unload method.

Invoking HPU within a work statement list

Before using HPU within a work statement list, be sure to enable HPU. The main HPU program (INZUTILB) needs to be authorized in the IKJTSOnn member of PARMLIB.

The Migrate function has a slightly different implementation than other functions, as the unload is performed before the work statement list is created, using regular JCL and not under the control of the ADBTEP2 program.

For functions other than MIGRATE, you decide to use HPU when you run the work statement list. On the Work Statement List Library panel (ADB2W1), enter the R line command to display the HPU Unload Prompt pop-up panel (ADB2WHPU) that indicates that an unload is being performed. At that time, you can decide whether to use HPU.

Restriction: The following restrictions apply to using HPU:
- Do not specify HPU if an object to be unloaded in the work statement has a security label column because the unload will fail.
- If the WSL includes an UNLOAD statement and a template substitution variable is part of the unload SYSREC template, HPU cannot be used. DB2 UNLOAD will be used instead, and the HPU Unload Prompt pop-up panel (ADB2WHPU) will not be displayed.

Because using HPU is determined at run time, all work statement lists are created using either UNLOAD or REORG UNLOAD EXTERNAL. You can select options R or U as the unload method when creating the work statement list. Selecting the H option does not specify that HPU will be used, but you can specify that you want to use HPU on the HPU Unload Prompt pop-up panel (ADB2WHPU) from ADB2W1.

You can port a work statement list from subsystem to subsystem. For example, if a work statement list is created on a subsystem that does not have HPU enabled, you can copy that work statement list to another subsystem that has HPU enabled.

If you do not select HPU at run time, the work statement list runs using the DB2 utility. Prior to submitting the work statement list jobs, you can choose between the DB2 utility and HPU.

Restriction: After the run is started, the unload method cannot be changed. For example, a job that fails using the DB2 UNLOAD utility cannot be restarted using HPU.

When an HPU job is being run using a work statement list, partitioned table spaces are unloaded by partition. The subsequent loading of the data is performed in parallel when possible; otherwise, the data sets are concatenated to form a single input stream.

Loads are performed serially in the following cases:
- When a table is loaded into a nonpartitioned table space
- When the number of partitions has changed
- When the partition key ranges have changed
- When an identity column appears in the partitioning index

Using HPU with MIGRATE and work statement lists

When migrating DB2 data, the Migrate Parameters panel (ADB28M) offers the option to specify an HPU unload.

You can specify that you want to unload the partitions in parallel. This option is ignored if you do not choose the HPU option. The JCL that is generated directly invokes DB2 HPU to complete the unload, as well as to create the work statement list. Because the work statement list does not contain an unload statement, no prompt is offered that asks whether HPU is required at run time. When the work statement list is run, the ADBTEP2 program automatically determines if the data was unloaded by partition and completes the appropriate steps to reload the data accordingly.

Note: You must set the parameter ULACCTRL=YES in the HPU PARMLIB, or the HPU job will not run correctly.

HPU settings

For the HPU job to run correctly, you must have the following parameter settings in the HPU PARMLIB:
- ULACCTRL=YES
- ULOPTNS=(LOADINDDN(YES))

Using HPU in a work statement list that is not created by DB2 Object Comparison Tool, ALTER, or ALT

All work statement lists that contain an UNLOAD or REORG UNLOAD EXTERNAL statement displays the HPU Unload Prompt pop-up panel (ADB2WHPU) at run time, provided that HPU is enabled.

The HPU support in DB2 Admin is primarily intended to be used for a work statement list that is created by one of the DB2 Admin or DB2 Object Comparison Tool functions. However, if HPU is selected at run time, any eligible unload is converted to run as an HPU unload. To be considered as an eligible unload, all of the following statements must be true:

- The UNLOAD statement, whether it be UNLOAD or REORG UNLOAD EXTERNAL, must have exactly one FROM TABLE clause, with no other keywords from the utilities FROM-TABLE-*spec*.
- The UNLOAD data set name must not exceed 38 characters. This restriction enables a suffix to be appended to the data set name that indicates the partition number.
- The DDNAME that is associated with the UNLOAD data set must be SYSREC.

Restriction: Do not code HPU syntax directly in a work statement list. Use only the DB2 utility format. When the ADBTEP2 program runs HPU on a partitioned table space, it always unloads each partition into a separate data set. For a work statement list that is not created using ALTER or DB2 Object Comparison Tool, you must ensure that subsequent handling of the output from the unload operation is managed appropriately.

How HPU reads the DB2 catalog

DB2 High Performance Unload can directly access the DB2 catalog.

DB2 Admin does not specify the options that apply to non-externalized updates to the catalog data in the DB2 buffer pools. You can provide this access by defining a default in the HPU PARMLIB member using one of the following options:

- Quiesce the catalog using option QUIESCECAT=YES
- Provide direct access without flushing the DB2 buffers using QUIESCECAT=NO. This can lead to failures.
- Specify that HPU uses DB2 to perform the catalog access using option SQLACCES=YES.

Recommendation: Whenever possible, use the last option listed in the previous list. (This option was provided in APAR PQ68392.)

Creating work statement lists manually

You can manually create or edit WSLs.

A benefit to manually creating a WSL is that you can use the WSL infrastructure to control related tasks. For example, if you want to run a heavy updating batch program, schedule an image copy, and RUNSTATS immediately after it, you could create a WSL containing these three tasks. The benefit is that the WSL is cloned and during execution the restart capability of ADBTEP2 is available.

Running work statement list entries in parallel

Within any WSL, you can edit the order (sequence) of the statements.

In addition, you can elect to run certain parts in parallel (where appropriate). Running jobs in parallel refers to creating multiple jobs that you or a scheduling system can run at the same time, instead of one after another. For example, you can run the unload jobs in parallel. Some of the input processes to the WSL (for example, from DB2 Object Comparison Tool) does this for you.

To run statement pairs in parallel, use a statement type of ADM and use the statements PARALLEL and ENDPARALLEL, and JOB and ENDJOB.

The PARALLEL and ENDPARALLEL statements signify the start and end points for jobs to be run in parallel. The JOB and ENDJOB card statements signify the start and end points of WSL statements for a particular job. You should have multiple JOB/ENDJOB pairs within a PARALLEL/ENDPARALLEL pair. WSL statements not included in a PARALLEL/ENDPARALLEL pair are placed in a separate job.

If you specify PARALLEL *name*, the members generated by RUN are suffixed by *xxxxn*, where *xxxx* is the user ID and *n* is the first character of *name*.

```
ADM PARALLEL UNLOAD
ADM JOB
tasks for job1

ADM ENDJOB
ADM JOB
tasks for job2

ADM ENDJOB
ADM ENDPARALLEL
serial tasks
```

This example results in three jobs. The first two jobs run concurrently and the third one runs when the first two are complete.

For multiple tables unload all the tables in parallel. When finished, run DDL to drop and redefine then run the loads in parallel.

The loads and unloads are run in parallel to increase performance. The DDL is done in one job to avoid DB2 locking or serialization problems.

Supplying input to the batch restart program (ADBTEP2)

The Batch Restart program (ADBTEP2) enables you to restart or resume the execution of an input stream at an intermediate point, in the event that any one of the statements in that stream should fail.

The process involves creating or updating a record in a checkpoint table each time that a COMMIT statement is encountered in the input stream. By using this table, execution can be resumed with the first statement following the last successful commit point before the failure, bypassing all prior successfully executed statements. Of course, before restarting after a failure, you must correct the condition that caused the failure.

In addition to SQL statements, you can supply DB2 commands, DB2 utilities, DB2 Admin support commands, and DSN commands as input to ADBTEP2.

The following types of input to ADBTEP2 are valid:

Remember: As with all ADBTEP2 commands, the semicolon delimiter (;) is required.

DB2 commands

The format is *-command*.

Example: -DIS GROUP

DSN Commands

The following DSN commands are supported:
- BIND
- DCLGEN
- FREE
- REBIND
- RUN

DB2 Utilities

The following DB2 utilities are supported:
- CHECK
- COPY
- COPYTOCOPY
- DIAGNOSE
- LOAD
- MERGECOPY
- MODIFY
- QUIESCE
- REBUILD
- RECOVER
- REORG
- REPAIR
- REPORT
- RUNSTATS
- STOSPACE
- UNLOAD

UTILFROM Utility

The DB2 UTILFROM utility is a *pseudo* utility that directs ADBTEP2 to execute the utility control statements that are contained in a data set. Only one utility can be contained within the data set so it is not possible to include RUNSTATS and LOAD in one UTILFROM. The purpose of the utility is to allow the LOAD control statements generated by UNLOAD, REORG UNLOAD, and HPU to be executed. Because UNLOAD does not produce all the control statements required (for example, SORTNUM), you must add them by using the ADD keyword.

The format of UTILFROM is UTILFROM *dsname* ADD(*additional control statements*).

Example:
```
UTILFROM ROYC.ROYCDOC1.CNTLC.PPP2
   ADD(SORTNUM 8 SORTDEVT SYSDA
     WORKDDN(UTLUT1,UTLOUT) ERRDDN(UTLERR)
     DISCARDDN(UTLDISC) MAPDDN(UTLMAP));
```

Functional comments

You can include the following functional comments:

--#SET ROWS_FETCH *n*

where *n* is a non-negative integer that indicates the maximum number of rows to be FETCHed for each subsequent SELECT statement. Use -1 to indicate that all rows should be fetched.

--#SET ROWS_OUT *n*
> where *n* is a non-negative integer that indicates the maximum number of rows to be output for each subsequent SELECT statement. Use -1 to indicate that all rows should be output.

--#SET TERMINATOR *n*
> where *n* is a one-byte character to be used to terminate the next SQL statement. Any character is valid, except blank, comma, single quotation, double quotation, underscore, and parentheses.

--#SET ACCEPT_RC (ON/OFF) *m n*
> where *m* or *n* is the SQLCODE that can be accepted for the SQL statements before the program stops. The maximum number of SQLCODE that can be listed is 5. Using --#SET ACCEPT_RC *m n* can accept SQLCODE *m* or *n* for the following single SQL statement. Using --#SET ACCEPT_RC ON *m n* can accept SQLCODE *m* or *n* for the following multiple SQL statements until the next --#SET ACCEPT_RC OFF occurs. If no SQLCODE is provided after --#SET ACCEPT_RC (ON/OFF), it means all SQLCODEs can be accepted.

--#SET MAXERRORS *n*
> where *n* is the number of DSN commands that can fail before the program stops. Use -1 to indicate that the program should tolerate an unlimited number of errors for DSN commands.

IBM reserves the right to use additional parameters in these functional comment statements. These parameters might be present in the statements that DB2 Admin generates for ADBTEP2. Do not modify these statements unless you are requested to do so by your IBM service representative.

REXX EXECS
> The format is *REXX execname parameters*
>
> *execname* can be the name of a CLIST. Programs are not supported. DB2 programs can be executed by using the DSN command RUN.

DB2 Admin support commands
> The following commands are considered DB2 Admin support commands. These commands are associated with (or support) primary commands that are located further down in the batch statement list. For example, the ALLOC command is used to allocate files for a program (the primary command). Support command processing is deferred until the primary command is encountered. Support commands must immediately precede their primary command.

ADBSYSIN
> Many programs, including ADBTEP2, use the filename (or DDNAME) SYSIN. ADBTEP2 uses SYSIN for the batch statement list; therefore, ADBSYSIN is used to identify the location of the input. The format is
> `ADBSYSIN COPYDD(`*ddname*`)`
>
> where *ddname* contains the SYSIN for the program following the ADBSYSIN.

ADBPAUSE
> You can use the ADBPAUSE statement to pause the current run of ADBTEP2 or ADBTEPA at a certain point. You can then restart ADBTEP2 and ADBTEPA at that point.

ALLOC
> A TSO ALLOCATE command is issued with the parameters provided.

ALLOC is intended to support programs only. It is not a valid support command for a DB2 utility (see TEMPLATE).

Example: `ALLOC DD(DATAI001) DS('ROYC.ROYCDOC1.UNLD.PPP1') SHR`

CHECKBEGIN and CHECKEND

The CHECKBEGIN and CHECKEND statements delimit a block of CHECK DATA commands. When CHECKEND is reached, DB2 Admin identifies the parent and children tables in RI relationships with the table spaces that are identified in the CHECK DATA commands within the block and generates CHECK DATA commands to clear these tables of any CHECK-pending status. Any TSODELETE commands before the CHECKEND are executed for all the generated CHECK DATA commands. Any TEMPLATE commands before the CHECKEND are supplied to the utility for all the generated CHECK DATA commands.

Example: In the following example, the second set of TSODELETE and TEMPLATE commands apply to the CHECK DATA commands that might be generated for the parent and descedent tables:

```
CHECKBEGIN;
TSODELETE 'JIMWC.EB12.CSUT1.T0001';
TSODELETE 'JIMWC.EB12.CSOUT.T0001';
TSODELETE 'JIMWC.EB12.CSERR.T0001';
TEMPLATE UTLUT1 DSN 'JIMWC.EB12.CSUT1.T0001'
        UNIT SYSDA;
TEMPLATE UTLOUT DSN 'JIMWC.EB12.CSOUT.T0001'
        UNIT SYSDA;
TEMPLATE UTLERR DSN 'JIMWC.EB12.CSERR.T0001'
        UNIT SYSDA;
CHECK DATA TABLESPACE DB2144.TS2144
  ERRDDN(UTLERR) WORKDDN(UTLUT1,UTLOUT)
  SORTDEVT SYSDA SORTNUM 4;
TSODELETE 'JIMWC.EB12.CSUT1.T0001';
TSODELETE 'JIMWC.EB12.CSOUT.T0001';
TSODELETE 'JIMWC.EB12.CSERR.T0001';
TEMPLATE UTLUT1 DSN 'JIMWC.EB12.CSUT1.T0001'
        UNIT SYSDA;
TEMPLATE UTLOUT DSN 'JIMWC.EB12.CSOUT.T0001'
        UNIT SYSDA;
TEMPLATE UTLERR DSN 'JIMWC.EB12.CSERR.T0001'
        UNIT SYSDA;
CHECKEND;
```

TEMPLATE

TEMPLATE is a utility support command. ADBTEP2 passes this command to the DB2 Utility processor. ADBTEP2 performs a partial simulation of the DB2 TEMPLATE function for TEMPLATE names that are not supported by DB2 (for example, SYSREC). The main difference between DB2 allocation of templates and the simulation is at failure, as the failure disposition is not honored. ADBTEP2 does not support utility wild cards.

TSODELETE

A TSO DELETE command is issued for the data set provided. If the DELETE fails, a DELETE NOSCRATCH is attempted. Processing continues even if TSODELETE is unsuccessful.

Chapter 16. Using the Batch Restart programs: ADBTEP2 and ADBTEPA

The Batch Restart program, ADBTEP2, and the Authorization Switching Program, ADBTEPA, are used with work statement lists and the Alter and Migrate DB2 data functions.

The Batch Restart program (ADBTEP2) enables you to restart or resume the execution of an input stream at an intermediate point, in the event that any one of the statements in that stream should fail. ADBTEPA allows user IDs that are not authorized to certain objects to re-create those objects if they are implicitly dropped.

Topics:

Introduction to ADBTEP2

The Batch Restart program (ADBTEP2) enables you to restart or resume the execution of an input stream at an intermediate point, in the event that any one of the statements in that stream should fail.

The process involves creating or updating a record in a checkpoint table each time that a COMMIT statement is encountered in the input stream. By using this table, execution can be resumed with the first statement following the last successful commit point before the failure, bypassing all prior successfully executed statements. Of course, before restarting after a failure, you must correct the condition that caused the failure.

In comparison, ADBTEP2 does not include all functions available in DB2 Admin Space Manager. For example, ADBTEP2 can support the changing of VCAT names for a table space or an index only when the VCAT names are defined within the same catalog structure.

Input types

ADBTEP2 can run the following elements from an input stream (SYSIN):
- SQL statements
- DB2 utilities
- DB2 commands
- DSN commands (including RUN)

- REXX EXECS or CLISTS

This input stream is referred to as a *batch statement list*.

Checkpoint table

ADBTEP2 is generally used in jobs that are generated by DB2 Admin, but it can also be used independently. The checkpoint table is a shared resource, and is named ADBCHKPT. You can determine the qualifier of this table by using the ADBTEP2 package associated with the plan that you are running (ADBTEP2 by default). ADBTEP2 adds and maintains a row in the checkpoint table. This row in the checkpoint table is referenced by a worklist name parameter that is supplied to ADBTEP2. The worklist name parameter is used in conjunction with the user ID of the submitter (to ensure uniqueness). The worklist name parameter is created when the JCL is generated by DB2 Admin functions and uses the work statement list name concatenated with an optional suffix.

The checkpoint table is updated at commit points to enable restarting. ADBTEP2 always performs implicit commits before and after performing functions other than SQL (for example, a DB2 utility). To issue a commit between SQL statements, add an SQL COMMIT statement. You can also instruct ADBTEP2 to commit after every statement by using the `commit_all` ADBOPT parameter.

Parameters passed to the ADBTEP2 program

When DB2 Admin generates the JCL to run ADBTEP2, parameters are generated automatically and are passed to ADBTEP2.

Parameters passed in the PARMS field of the DB2 RUN statement

The following parameters are generated automatically and are passed to the ADBTEP2 program in the PARMS field of the DB2 RUN statement:

MAXE(-1, 0, 1-99)
    Specifies the number of DSN commands that can fail before the batch job is terminated:

    -1   All errors are ignored. The batch job is not stopped for any error.

    0   No errors are allowed. The batch job is stopped on the first error.

    1-99
        The specified number of errors are ignored. The batch job is stopped on the next DSN command that fails. For example, if you specify 5, the batch job is stopped when the sixth DSN command fails.

    Any failing DSN command statements that are ignored are skipped and are written to the ADBHOLD table. When the job ends, if any DSN commands have failed, the restart action field in the checkpoint table contains an 'H' to indicate that there are held records. When RESTART(YES) is specified, the held records are reprocessed if the batch job ended with a return code of 0; otherwise, the job is restarted from the last recorded commit point. When RESTART(NO) is specified, the held records are purged and the job is restarted from the beginning.

RESTART

RESTART(NO)
Indicates that ADBTEP2 does not perform a restart, and execution starts with the first command. The WORKLIST() parameter must be used with this option, and ADBTEP2 updates the checkpoint table. A subsequent restart can be performed by using RESTART(YES).

RESTART(YES)
Indicates that the job is to be restarted from the last recorded commit point prior to a failure. RESTART(YES) is the default. If RESTART(YES) is specified or used as a default, you must also provide the WORKLIST() parameter. When execution begins, ADBTEP2 searches for a checkpoint row in the checkpoint table and repositions within the input, skipping over committed commands.

RESTART(YES) causes a very basic check to be done. RESTART(YES) checks that the last command type held in the checkpoint record matches the command type to be attempted at restart. This check is performed to prevent an accidental reuse of a checkpoint against a completely different WSL.

Recommendation: Exercise caution when editing the input stream between ADBTEP2 failures. If the checkpoint record is not found, ADBTEP2 starts with the first command in the input stream.

RESTART(FORCE)
As with RESTART(YES), RESTART(FORCE) restarts at the last commit point prior to a failure. You must also provide the WORKLIST() parameter. However, the basic check done in RESTART(YES) is not done in RESTART(FORCE). Because the basic check is not done, the restart point might be unintended and the results might be unpredictable.

If the COMMAND_RESTART column in the ADBCHKPT table has a value of 'S' upon the restart processing, the check for the checkpoint record is not performed. And, if the checkpoint dialog **Skip-Next** line command is used, the check is not performed

WORKLIST(*extended-name*)
extended-name is a unique identifier that is used in conjunction with the user ID of the submitter to provide the key for the checkpoint record. The full format of *extended-name* is *name.suffix*, where *name* includes 1-8 alphanumeric characters, and *suffix* includes 1-8 alphanumeric characters. The separator must be a period (.). The suffix is optional, but if the suffix is omitted, the separator must also be omitted.

For jobs that DB2 Admin generates, *name* is the same as the work statement list.

Examples:

WORKLIST(TEST1)
Simple worklist name

WORKLIST(TEST2.N00005)
Worklist including suffix

The following parameters are passed to ADBTEP2 and are used to control non-restart functions:

ALIGN

ALIGN(MID)
>
> Aligns output from the program to the center of the page. This is the default.

ALIGN(LHS)
>
> Aligns output from the program to the left-hand side of the page.

MIXED

MIXED
>
> Indicates that the input stream can contain data in a combination of SBCS and DBCS formats.

NOMIXED
>
> Indicates that the input stream will contain data in SBCS format only. This is the default.

PCACT
>
> Specifies the action to take when the job is to recover a change made through Change Management and pending changes exist that affect the same objects or related objects as the change.

PCACT(CANCEL)
>
> Indicates that the recover job will not be run.

PCACT(SUPERSEDE)
>
> Indicates that the recover job will be run. The recover change supersedes the pending changes, and the pending changes are set to DEFINED status.

SQLTERM(*c*)
>
> *c* defines the character that terminates an SQL statement. The default termination character is the semicolon (;).

SSID(name)
>
> A subsystem or group attachment name to be used for running non-SQL commands or functions. This name should be the same as that used in the DSN SYSTEM(*xxxx*), which is used ahead of the RUN command that invokes ADBTEP2. This parameter is required if any non-SQL DB2 function is included in the input stream (for example, a DSN command).

Parameters passed under the DD name of ADBTEPIN

The following parameters are generated automatically and are passed to the ADBTEP2 program in a data set with a DD name of ADBTEPIN:

Advisory Auto Rebuild
>
> The Advisory Auto Rebuild parameter determines if the Batch Restart Program initiates a REBUILD of an index when an object is in the ARBDP state.
>
> - YES - A REBUILD is attempted. However, if the parameter **Run REORG/REBUILD** was specified as 'A - All relevant' to generate an explicit REBUILD during the change flow, then the value NO is passed to ADBTEP2. The NO value prevents an automatic REBUILD that duplicates the explicit REBUILD.
> - NO - A REBUILD is not attempted.
>
>   No is the default.
>
> For more information about the rebuild-pending states, see the *DB2 V10 Utilities Guide*.

Tip: To prevent the ADBTEP2 program from scheduling any automatic REBUILDs, you must set Auto Rebuild, Advisory Auto Rebuild and Auto Reorg/Rebuild after STOGROUP change parameters all to No.

Advisory Auto Reorg

The Advisory Auto Reorg parameter determines if the Batch Restart Program initiates a REORG of a table space when an object is in the AREOR,AREO* state.

- YES - A REORG is attempted. However, if the parameter **Run REORG/REBUILD** was specified as 'A - All relevant' to generate an explicit REORG during the change flow, then the value NO is passed to ADBTEP2. The NO value prevents an automatic REORG that duplicates the explicit REORG.
- NO - A REORG is not attempted.

  No is the default.

For more information about the rebuild-pending states, see the *DB2 V10 Utilities Guide*.

Tip: To prevent the ADBTEP2 program from scheduling any automatic REORGs, you must set Auto Reorg, Advisory Auto Reorg and Auto Reorg/Rebuild after STOGROUP change parameters all to No.

Auto Rebuild

The Auto Rebuild parameter determines if the Batch Restart Program initiates a REBUILD of an index when an object is in the RPDB. RPDB*, or PSRBD state.

- YES - A REBUILD is attempted. However, if the parameter **Run REORG/REBUILD** was specified as 'M - Mandatory' or 'A - All relevant' to generate an explicit REBUILD during the change flow, then the value NO is passed to ADBTEP2. The NO value prevents an automatic REBUILD that duplicates the explicit REBUILD.
- NO - A REBUILD is not attempted.

  No is the default.

For more information about the rebuild-pending states, see the *DB2 V10 Utilities Guide*.

Tip: To prevent the ADBTEP2 program from scheduling any automatic REBUILDs, you must set Auto Rebuild, Advisory Auto Rebuild and Auto Reorg/Rebuild after STOGROUP change parameters all to No.

Auto Reorg

The Auto Reorg parameter determines if the Batch Restart Program initiates a REORG of a table space when an object is in the REORP state.

- YES - A REORG is attempted. However, if the parameter **Run REORG/REBUILD** was specified as 'M - Mandatory' or 'A - All relevant' to generate an explicit REORG during the change flow, then the value NO is passed to ADBTEP2. The NO value prevents an automatic REORG that duplicates the explicit REORG.
- NO - A REORG is not attempted.

  No is the default.

For more information about the reorg-pending states, see the *DB2 V10 Utilities Guide*.

Tip: To prevent the ADBTEP2 program from scheduling any automatic REORGs, you must set Auto Reorg, Advisory Auto Reorg and Auto Reorg/Rebuild after STOGROUP change parameters all to No.

Auto Reorg/Rebuild after STOGROUP change

The Auto Reorg/Rebuild after STOGROUP change parameter determines if the Batch Restart Program initiates a REORG or REBUILD after ALTER STOGROUP statement is executed for the table space or index.

- YES - A REORG or REBUILD is attempted. However, if the parameter **Run REORG/REBUILD** was specified as 'A - All relevant' to generate an explicit REORG or REBUILD during the change flow, then the value NO is passed to ADBTEP2. The NO value prevents an automatic REORG or REBUILD that duplicates the explicit one.
- NO - A REORG or REBUILD is not attempted.

  No is the default.

Tip: To prevent the ADBTEP2 program from scheduling any automatic REORGs, you must set Auto Reorg, Advisory Auto Reorg and Auto Reorg/Rebuild after STOGROUP change parameters all to No. To prevent the ADBTEP2 program from scheduling any automatic REBUILDs, you must set Auto Rebuild, Advisory Auto Rebuild and Auto Reorg/Rebuild after STOGROUP change parameters all to No.

Autocheck

Certain SQL or utility operations can place an object into check-pending state. If you set the Autocheck (AC) parameter value to YES, ADBTEP2 tracks the following statements and processes that can place an object in check-pending state. If a statement or process is encountered, ADBTEP2 performs an automatic CHECK DATA to remove the check-pending state. The default value for AC is NO.

ADBTEP2 tracks the following statements:

- ALTER TABLE ... ADD FOREIGN KEY
- ALTER TABLE .... ADD CONSTRAINT
- LOAD REPLACE
- LOAD ENFORCE(NO)
- RECOVER PIT

ADBTEP2 tracks the following processes:

- COPY utility - perform auto-check prior to COPY
- CHECK DATA utility – perform auto-check after CHECKEND
- A final auto-check at the end of the SYSIN input stream

Restriction: DB2 Admin builds the CHECK DATA statement and all CHECK parameters used during auto-check processing. You cannot specify any other parameters.

BINDERROR(MAXE, SAVE, IGNORE)

Specifies how BIND or REBIND errors that are processed by ADBTEP2 are to be handled.

MAXE

The failing BIND or REBIND command is written to the ADBHOLD table. The value that is specified for the MAXE parameter determines whether ADBTEP2 continues to process the input stream:

- If MAXE(0) is specified or if MAXE() is omitted, processing stops.

- If MAXE(-1) is specified, processing continues.
- If a value greater than 0 is specified for MAXE, the MAXE counter is incremented by 1, and processing stops if the number of errors has exceeded the maximum number of allowed failures.

SAVE

The failing BIND or REBIND command is written to the ADBHOLD table. ADBTEP2 continues to process the input stream.

IGNORE

The failing BIND or REBIND command is ignored and is not written to the ADBHOLD table. ADBTEP2 continues to process the input stream.

DB2 Pending Changes options (DB2 Version 10 New Function mode only):

The Check at DROP parameter controls if a check is made to avoid losing any DB2 pending changes as part of the DROP action.

- YES - The DROP is not performed if a DB2 pending change exists.
- NO - The DROP is performed without checking for pending changes.

Log DIAG

Controls whether diagnostic messages are written to the ADBDIAG file.

Yes

Messages are written to this file, which IBM Software Support can use to determine the cause of a failure.

No Messages are not written.

LOAD Summary Report

Controls if the LOAD summary report is produced as part of the ADBTEP2 run.

Yes

The LOAD Summary report is produced.

No The LOAD Summary report is not produced.

LOB/XML IC Unload

Controls the behavior of UNLOAD TABLESPACE statements if an unload from an image copy of a table space is requested and a table in that table space contains a LOB or XML column.

E The ADBTEP2 program should end with an error.

U An unload of the base object should be performed instead.

Maxerrors

The number of DSN commands that can fail before the batch restart job ADBTEP2 is stopped:

-1 All errors are ignored. The batch job is not stopped for any error.

0 No errors are allowed. The batch job is stopped on the first error.

1-99

The specified number of errors are ignored. The batch job is stopped on the next DSN command that fails. For example, if you specify 5, the batch job is stopped when the sixth DSN command fails.

Any failing DSN commands that are ignored are skipped and are written to the ADBHOLD table. When the job ends, if any DSN commands have failed,

the restart action field in the checkpoint table indicates that there are held records. Depending on the restart option, the held records are reprocessed when the job is restarted.

Missing IC Unload

Controls the behavior of UNLOAD TABLESPACE statements if an unload from an image copy of a table space is requested and no image copy can be found.

E The ADBTEP2 program should end with an error.

U An unload of the base object should be performed instead.

SQLFORMAT

Specifies how ADBTEP2 pre-processes SQL statements before passing them to DB2. Currently, ADBTEP2 only supports option SQLCOMNT.

SQLCOMNT

This mode is suitable for all SQL, but it is intended primarily for SQL procedural language processing. When this option is in effect, ADBTEP2 does not discard SQL comments, and automatically terminates each SQL comment with a line feed character (hex 25) unless the comment is already terminated by one or more line-formatting characters. Note that the option SQLFORMAT = 'SQLCOMNT' must be added manually to ADBTEPIN DD.

Overriding WSL restart parameters

You can override the parameters that the ADBTEP2 program uses when performing a restart.

1. Ensure the Work Statement List Library panel is open.
2. Type the R line command next to a WSL a checkpoint.

   Look for Y under the Restart column.
3. On the Specify Restart Information panel, type the V line command to edit the restart information.
4. Override the parameters. You can override the following parameters:

   Decfloat Rounding Mode

   Specifies the system default action that is used for rounding decimal floating point values.

   Path Specifies the SQL path used when resolving unqualified function names, procedure names, data type names, and module object names in dynamically prepared SQL statements.

   Precision

   Specifies the CURRENT PRECISION.

   Routine Version

   Assigns a value to the CURRENT ROUTINE VERSION special register.

   Rules Specifies the CURRENT RULES.

   SCHEMA

   Specifies the current schema special register to use at the restart point.

   Server Specifies the location name of the current server.

   SQLID

   Specifies the current SQLID.

Use of a REXX routine with the ADBTEP2 program

A REXX routine can provide statements to ADBTEP2 for processing.

You call a REXX routine from ADBTEP2:

```
REXX %<name> [parm];
```

To provide input to ADBTEP2, you use a functional comment before the syntax. The comment informs ADBTEP2 that the REXX routine is providing information for ADBTEP2 to process. You can provide input for the following functions:

- User statements that are in a form that can be processed by ADBTEP2, for example SQL statements, DB2 commands, or DSN commands.
- Iterative processing

You end the input statements with a semi-colon (;).

You must issue DSNREXX DISCONNECT in the REXX routine before you can use any command that requires ADBTEP2 to connect to DB2.

You can provide information to ADBTEP2 through the user (USERINFO) and utility information (UTILINFO) functions. You can specify a tolerance threshold for utility errors. And you can allocate output from REXX-provided statements processed by ADBTEP2 to a USRPRINT file.

User input

The user input function enables the REXX routine to provide statements on the REXX data stack to ADBTEP2.

The syntax is as follows:

```
--#GET INPUT FROM STACK
REXX %<name> [parms];
```

The return code from the REXX routine specifies the action that ADBTEP2 takes:

RC=0 Statements are present on the data stack. The REXX routine writes statements onto the data stack for ADBTEP2 to process. ADBTEP2 pulls the statements from the data stack and processes statements until all statements are processed or until an error occurs.

RC=4 No statements are present on the data stack.

RC<>0, RC<>4
An error occurred and ADBTEP2 is directed to end processing.

Iterative input

The iterative input function prompts ADBTEP2 to repeat invocation of a REXX routine.

The syntax is as follows:

```
--#GET INPUT FROM STACK WITH ITERATION
REXX %<name> [parms];
```

The return code from the REXX routine specifies the action that ADBTEP2 takes:

RC=0 Statements are present on the data stack. ADBTEP2 pulls the statements from the data stack and processes statements until all statements are processed or until an error occurs.

RC=4 No statements are present on the data stack.

Until RC=4
ADBTEP2 reinvokes the REXX routine to get more statements until the REXX routine ends with RC=4.

RC<>0, RC<>4
An error occurred and ADBTEP2 is directed to end processing.

User information

The user information function enables the REXX routine to provide information for iterative REXX calls. The user information function is for iterative input only.

The syntax that prompts ADBTEP2 to process a REXX statement is as follows:

```
USERINFO <string>;
```

The user information statement enables the REXX routine to identify the work that is passed to ADBTEP2. ADBTEP2 writes the statement back to the data stack when the REXX routine is invoked the next time, and only if the call is part of iterative input processing.

The following example shows how you can call a REXX routine that passes a USERINFO string to ADBTEP2 and directs ADBTEP2 to run statistics on a tablespace:

```
/* rexx */
arg exitrc
queue "USERINFO RUN RUNSTATS ON A TABLE SPACE;
queue "RUNSTATS TABLESPACE ADBDCHG.ADBSPF1",
      "   INDEX",
      "   (",
      "    ALL",
      "   )",
      "  SHRLEVEL CHANGE;"
queue ""
exit exitrc
```

Utility Information

The utility information function enables a REXX routine to provide utility identification information, through ADBTEP2, to DB2.

The syntax that prompts ADBTEP2 to receive utility identification information from a REXX routine and to pass the information to DB2 is as follows:

```
UTILINFO [SYSTEM<ssid>,][UID=<utility-id>],[UTPROC=<utproc-string>];
```

The UTILINFO statement must precede the utility statements to which they apply. Multiple parameters must be separated by a comma. The statement must end with a semi-colon (;).

When parameters are not provided in the REXX statement, the default action is for ADBTEP2 to use parameters that are passed to ADBTEP2:

SYSTEM
> The value of the SSID() parameter that is passed to ADBTEP2 and then is passed to DB2

UID   The value of the WORKLIST() parameter that is passed to ADBTEP2 and then is passed to DB2

UTPROC
> blank. Passes the supplied JCL procedure, if any, to DB2.

You can call a REXX routine that directs ADBTEP2 to pass DB2 utility parameters, SYSTEM and UID, to DB2. In the following example, the system name and utility ID are passed to ADBTEP2, and then ADBTEP2 runs the RUNSTATS utility:

```
/* rexx */
arg exitrc
queue "UTILINFO SYSTEM='DSNX',UID='VNDR2';"
queue "RUNSTATS TABLESPACE ADBDCHG.ADBSPF1",
      "  INDEX",
      "   (",
      "    ALL",
      "   )",
      "  SHRLEVEL CHANGE;"
queue ""
exit 0
```

Tolerance threshold for DB2 utility command error return codes

The tolerance threshold enables you to specify the error return code number, for a DB2 utility command error, to be tolerated during the processing of REXX statements. When the specified threshold is exceeded, ADBTEP2 stops processing.

The syntax that specifies the return code of errors that are tolerated is as follows:
```
--#SET TOLUTILERR n
```

The value of *n* is the return code number and must be an integer between 4 to 32767. When processing iterative statements in a REXX routine, the REXX routine, that includes DB2 utility commands, iterates until a return code that is beyond the threshold is encountered or until ADPTEP2 completes execution.

The following example shows that you specify return code tolerance before you specify a user input statement:
```
--#SET TOLUTILERR 7
--#GET      INPUT    FROM STACK WITH ITERATION
REXX T2IN2 0;
```

In the example, if the return code for a DB2 utility command error exceeds the value 7, ADPTEP2 stops processing.

User Print

The user print function enables you to send output from REXX statements processed by ADBTEP2 to a USRPRINT file.

You can specify that DB2 output from REXX-provided statements be written to a USRPRINT file. A USRPRINT file contains output only from DB2. USRPRINT is processed only when the DD statements of USRPRINT is provided. Alternatively, a SYSPRINT file contains all output from DB2 and DB2 Administration tool.

In order to use USRPRINT, the following requirements must be met:
- SYSPRINT and USRPRINT must be preallocated.
- SYSPRINT must be allocated as a non-spool dataset with DISP option as MOD.
- USRPRINT must use the same dataset attributes except DISP option.

You do not use a REXX statement. You use SYSPRINT and USRPRINT DD
statements in JCL to allocate the data sets:

```
//SYSPRINT DD DSN=<your data set>,
//           DISP=(MOD,CATLG,CATLG),
//           SPACE=(TRK,(10,10,0)),LRECL=137,RECFM=VB,BLKSIZE=141
//USRPRINT DD DSN=<your data set>,
//           DISP=(NEW,CATLG,CATLG),
//           SPACE=(TRK,(10,10,0)),LRECL=137,RECFM=VB,BLKSIZE=141,
//           VOL=SER=<volume name>
```

Data sets that the Batch Restart Program (ADBTEP2) uses

The ADBTEP2 program uses several data sets during its operation.

The following table lists the data sets that the ADBTEP2 program uses. The table
lists the DD name that is used to identify the data set and a description of the data
set. All of these data sets are required.Include statements in your JCL for each
required data set and any optional data sets that you want to use.

Table 15. Data sets that ADBTEP2 uses

Data set	Description
SYSIN	Input data set that contains the input stream or batch statement list, which is supplied at run time to the Batch Restart Program.
SYSPRINT	Output data set for messages. When the REXX user input feature is used, the data set must be allocated with the MOD as DISP option. The dataset must not be a spool file, for example, USRPRINT must be defined.
SYSEXEC	Input data set that contains the Admin Tool EXECs
SYSTSPRT	Input data set that is used to control the output from your background job. By specifying different operands on this statement, you can have the output listed on a system printer, placed in a specified data set for later use, or held in a work data set, so you can look at it using the OUTPUT command.
MSGLIB	Data set that contains the IBM Language Environment® (LE) messages
USRPRINT	Output message data set for DB2 messages. Used when REXX user input feature is used, the data set must be pre-allocated.

Using ADBTEP2

You use a sample job that is generated during install time through the Tools Customizer to run ADBTEP2, the Batch Restart program. The generated job is located in the Product Customization Library.

Prerequisite: ADBTEPA is used only if the auth-switching function is enabled.

You must modify this job to conform to the conventions established in your installation and to provide the input data stream for execution (also referred to as the *batch statement list*). The names of job cards, data sets, plans, and subsystems are site specific. The Product Customization Library name is also site-specific.

The batch statement list can be specified inline, as a sequential data set, or as a member of a partitioned data set. It should contain all of the SQL statements, DB2 commands, utility control statements, and other valid statements that you want to process in a single execution. Within this series of statements, be sure to separate logical tasks or units of work with a COMMIT statement. These denote the points at which a failed execution can be restarted. Non-SQL functions have implicit commits, both before and after them.

Because all ADBTEP2 jobs are restartable, it is recommended that the `worklist` parameter is specified and provides a unique name. The `RESTART` parameter can be set to either `YES` or `NO` or used as default (`YES`), depending on whether the submission of the job is required to restart. ADBTEP2 is restartable regardless of the RESTART option. A job that is run with `RESTART(NO)`, can be resubmitted with `RESTART(YES)` in the event of a failure. When you have specified the parameters, submit the JCL for execution.

If the execution completes successfully, nothing more needs to be done. Upon successful completion, both ADBTEP2 and ADBTEPA delete the checkpoint record.

If the execution is unsuccessful, examine the output to determine the reason for the failure. Correct the error and resubmit the job.

Dialog support for batch job checkpoint table

To display and manage the checkpoint table (ADBCHKPT) associated with batch jobs running ADBTEP2, use the 2B–Display/Manage Batch Checkpoint Table option on the DB2 System Administration panel (ADB2Z).

For each active batch job running ADBTEP2 and for jobs running ADBTEP2 that have terminated because of an error in the input stream, a record of that execution is present in the checkpoint table. Select option 1, Display Checkpoint Records, from the Manage Batch Job Checkpoint Table panel to see those records, terminate an active ADBTEP2 job, update or delete the record of an abnormally terminated job, or insert a new checkpoint record.

Important: A new checkpoint record is only inserted to replace one that was deleted accidentally.
In addition, you can instruct ADBTEP2 to skip to the next commit using the N line command (skip-next).

Select option 2, Display Checkpoint Table Status, to obtain information about the checkpoint table itself, and issue any requests against the table, such as GRANT or REVOKE, that are supported by DB2 Admin.

The ADBTEP2 summary report

You might want a summary report of all activity at the end of or during large or complex work statement list (WLS) runs. This report will enable you to quickly spot any object or data availability issues. The ADBTEP2 summary report appears (and grows) while any ADBTEP2 job is running, not just WSLs.

The report can be examined in SDSF, under the ADBRPTSM DD. A sample report is shown in the following figure.

```
13:13.31 DB2 Administration Tool - 2009-09-17 Summary Report for L655527D
13:13.31
13:13.31 Ret Code Action   Object
13:13.31 ======== ======== =======================================================
13:13.32        0 UNLOAD   TABLESPACE DBADKK01.TSADKK01 FROM TABLE "VNDDHG"."TB
                           89740"
13:13.34        0 DROP     TABLESPACE "DBADKK01"."TSADKK01"
13:13.37      562 GRANT    USE OF STOGROUP SYSDEFLT TO USRT001
13:13.37      562 GRANT    USE OF STOGROUP SYSDEFLT TO "PUBLIC"
13:13.39        0 CREATE   TABLESPACE TSADKK01
13:13.39        0 CREATE   TABLE VNDDHG.RN89740
13:13.39      562 GRANT    USE OF STOGROUP SYSDEFLT TO "PUBLIC"
13:13.39        0 CREATE   TABLESPACE TSADKK01
13:13.39        0 CREATE   TABLE VNDDHG.RN89740
13:13.42        0 CREATE   UNIQUE INDEX "VNDDHG"."D7762_INDEX" ON "VNDDHG"."RN8
                           9740"
13:13.42        0 CREATE   UNIQUE INDEX VNDDHG.D7762_INDEX1 ON VNDDHG.RN89740
13:13.42        0 CREATE   VIEW VNDDHG.VW_TEACHER
13:13.42     -204 DROP     TRIGGER VNDDHG.INSOF_VIEW_TRIG01
13:13.42        0 CREATE   TRIGGER VNDDHG.INSOF_VIEW_TRIG01
13:13.44        4 UTILFROM VNDDHG.L655527D.CNC.T001
13:13.45        0 ALTER    TABLE "VNDDHG". "RN89740" ALTER COLUMN "TEACHER_ID"
                           SET GENERATED ALWAYS
13:13.45
13:13.45          End of Summary Report
```

Figure 249. ADBTEP2 summary report

Restarting an ADBTEP2 job

When ADBTEP2 runs, it checks to see if a record exists within the checkpoint table that matches the `worklist` parameter for the user ID that submitted the job.

If a record does not exist, ADBTEP2 creates it and starts with the first statement in the batch statement list. If a record exists, ADBTEP2 proceeds based on the `RESTART` parameter. When `RESTART(NO)` is specified, ADBTEP2 starts with the first statement in the batch statement list. When either no `RESTART` parameter is provided or `RESTART(YES)` is specified, ADBTEP2 repositions itself within the batch statement list and resumes processing.

ADBTEP2 has a simple restart capability. When the failing statement is SQL, a restart occurs at the last commit point prior to the failing SQL statement, which can be either an SQL COMMIT statement or an implicit commit that is performed while successfully completing a non-SQL function, such as a DB2 command.

Tip: It is important to avoid causing ADBTEP2 to reposition incorrectly when editing the batch statement list between runs. If the only change you require is to skip to the next commit instruction, use the N (skip-next) line command instead of editing the input to ADBTEP2. For an example of using the N (skip-next) line command, see the following figure.

If the failing statement is not an SQL statement, ADBTEP2 repositions to this statement. It is possible, although not likely, for the job to fail after executing non-SQL statements and before ADBTEP2 can update and commit the checkpoint record. In this case, ADBTEP2 positions on this non-SQL statement. Non-SQL statements cannot be rolled back if a failure occurs during ADBTEP2 checkpoint/commit. If you determine that the non-SQL statement completed, you can instruct ADBTEP2 to skip this statement on restart by using the N (skip-next) line command. ADBTEP2 reports the successful implicit commits that it performs before and after non-SQL statements. You can also determine whether ADBTEP2 failed on non-SQL statements by viewing the checkpoint record: the Restart Command field is blank if an SQL COMMIT was the last commit or if the last commit was an implicit commit as a result of completing a non-SQL statement. If the last commit was an implicit commit ahead of non-SQL statements, the Restart Command field is set to the type of non-SQL statement (for example, -STA).

If ADBTEP2 determines that a utility was running at the time of failure, ADBTEP2 obtains information from DB2 (if the utility is known to DB2) and restarts accordingly.

The following figure illustrates the checkpoint for the job with worklist DOC1. Because the Restart Command field is blank, we can determine that the last instruction performed was either an SQL COMMIT or a non-SQL statement that completed with an implicit commit. If we issue an N (skip-next) line command, Figure 251 on page 402 is displayed. The checkpoint number has been increased by one.

```
 DB2 Admin ---------- DB2X Display Batch Job Checkpoint Table - Row 1 to 4 of 4
 Command ===>


                                                      DB2 System: DB2X
                                                      DB2 SQL ID: ISTJE
 Checkpoint Table:  ADBB10.ADBCHKPT

 Line commands:
  D - Delete/Terminate  I - Insert  U - Update  N - Skip-Next
                                             Commit      Restart      Restart
 S Userid   Worklist Suffix  Time            Number      Command      Action
   *        *        *       *                           *            *
 - -------- -------- -------- --------------> ----------- ----------- ------
 n ROYC     DOC1              2002-07-18-16.06          4
   VNDBRON  RI03              2002-07-10-16.19          2
   VNDOJFK  OBJCMP            2002-06-26-16.54          1
   VNDROTH  AAA               2002-06-26-07.36          1 COPY        C
 ****************************** END OF DB2 DATA ******************************
```

Figure 250. Display Batch Job Checkpoint Table panel (ADB2Z2B1) – using the Skip-Next line command

```
DB2 Admin ---------- DB2X Display Batch Job Checkpoint Table - Row 1 to 4 of 4
Command ===>

                                                    DB2 System: DB2X
                                                    DB2 SQL ID: ISTJE
Checkpoint Table:  .ADBCHKPT

Line commands:
 D - Delete/Terminate  I - Insert  U - Update  N - Skip-Next
                                          Commit      Restart      Restart
S Userid   Worklist Suffix   Time          Number      Command      Action
  *        *        *        *                         *            *
- -------- -------- -------- ---------------- ----------- ----------- ------
  ROYC     DOC1              2002-07-18-16.06          5 UNKNOWN     N
  VNDBRON  RI03              2002-07-10-16.19          2
  VNDOJFK  OBJCMP            2002-06-26-16.54          1
  VNDROTH  AAA               2002-06-26-07.36          1 COPY        C
***************************** END OF DB2 DATA *******************************
```

Figure 251. Display Batch Job Checkpoint Table panel (ADB2Z2B1) – result of the Skip-Next line command

In Figure 252 on page 403, DOC2 has a Restart Command value that indicates that a COPY statement failed. The value in the Restart Action field determines the action to occur when ADBTEP2 repositions. For utilities, the value can be:

C   Restart current (ADBTEP2 default)

P   Restart phase

R   Restart from the beginning of the utility

S   Skip running the utility

The value in the Restart Action field can also be 'H', which indicates that the ADBHOLD table contains failed DSN commands. These failed DSN commands can be reprocessed when the job is restarted with RESTART(YES).

The U line command (Update) on this panel can be used to change the restart option for utilities. For example, you can change the C to an R. For non-SQL statements, only the options S (skip) and R (rerun or reissue) are valid.

Figure 253 on page 403 shows the result of using the N (skip-next) line command against DOC2. The restart command is now S and the commit number has not been increased. The Restart Command still displays the original type of the failing command, in this case COPY, as opposed to Figure 251, which shows the command as UNKNOWN.

```
DB2 Admin ---------- DB2X Display Batch Job Checkpoint Table - Row 1 to 4 of 4
Command ===>

                                                DB2 System: DB2X
                                                DB2 SQL ID: ISTJE
Checkpoint Table:  .ADBCHKPT

Line commands:
 D - Delete/Terminate  I - Insert  U - Update  N - Skip-Next
                                       Commit      Restart     Restart
S Userid   Worklist Suffix   Time      Number      Command     Action
  *        *        *        *                     *           *
- -------- -------- -------- ---------------- ----------- ----------- ------
n ROYC     DOC2              2002-07-18-16.16       5 COPY        C
  VNDBRON  RI03              2002-07-10-16.19       2
  VNDOJFK  OBJCMP            2002-06-26-16.54       1
  VNDROTH  AAA               2002-06-26-07.36       1 COPY        C
***************************** END OF DB2 DATA ******************************
```

Figure 252. Display Batch Job Checkpoint Table panel (ADB2Z2B1) – reissuing the Skip-Next line command

```
DB2 Admin ---------- DB2X Display Batch Job Checkpoint Table - Row 1 to 4 of 4
Command ===>

                                                DB2 System: DB2X
                                                DB2 SQL ID: ISTJE
Checkpoint Table:  .ADBCHKPT

Line commands:
 D - Delete/Terminate  I - Insert  U - Update  N - Skip-Next
                                       Commit      Restart     Restart
S Userid   Worklist Suffix   Time      Number      Command     Action
  *        *        *        *                     *           *
- -------- -------- -------- ---------------- ----------- ----------- ------
  ROYC     DOC2              2002-07-18-16.16       5 COPY        S
  VNDBRON  RI03              2002-07-10-16.19       2
  VNDOJFK  OBJCMP            2002-06-26-16.54       1
  VNDROTH  AAA               2002-06-26-07.36       1 COPY        C
***************************** END OF DB2 DATA ******************************
```

Figure 253. Display Batch Job Checkpoint Table panel (ADB2Z2B1) – result of reissuing the Skip-Next line command

Using ADBTEP2 with LOBs

If the UNLOAD statement is preceded with a LOB template, the UNLOAD statement input is modified by ADBTEP2 before it is passed to DB2 or High Performance Unload (HPU) so ADBTEP2 can unload LOB columns.

These modifications might be obvious only by examining the job log (SDSF output). The following example is a sample job log that shows JCL that is modified by ADPTEP2.

```
****************************** Top of Data ******************************
//SMITHSD JOB (SMITHS,X,090,IE1A),'DB2 UTILITY',
//*        RESTART=STEPNAME, <== FOR RESTART REMOVE * AND ENTER STEP NAME
//          REGION=0M,NOTIFY=SMITHS,
//          MSGCLASS=H,
//          CLASS=A
//*
/*JOBPARM S=SY4A
//*
//*
//************************************************************************
//*
//* DB2 BATCH MONITOR
//*
//* DB2 ADMIN GENERATED BATCH JOB.
//*


//************************************************************ADB2WL4**
//DB2B EXEC PGM=IKJEFT01,DYNAMNBR=100
//STEPLIB  DD DISP=SHR,DSN=DSN810.SDSNEXIT
//         DD DISP=SHR,DSN=DSN810.SDSNLOAD
//MSGLIB   DD DISP=SHR,DSN=ADBB10.SADBLLIB
//         DD DISP=SHR,DSN=GOCB10.SGOCLLIB
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//UTPRINT  DD SYSOUT=*
//RNPRIN01 DD SYSOUT=*
//ADBDIAG  DD SYSOUT=*
//SYSTSIN  DD *
 DSN SYSTEM(DB8A)
 RUN  PROGRAM(ADBTEP2) PLAN(CMDBKAT) -
   LIB('ADBB10.SADBLLIB')  -
   PARMS('/WORKLIST(TB2LOBS) SSID(DB2X)  -
 RESTART(YES)')
 END
//SYSIN    DD *
-- EDITED BY SMITHS  ON 2007/09/30 AT 02:28
-- EDITED BY SMITHS  ON 2007/09/30 AT 02:10
-- Created by SMITHS on 2007/09/30 at 02:06:58.45
TSODELETE 'SMITHS.DB2X.CNTL.LOB2DB.KAV2TS';
TEMPLATE UTLPUNCH DSN 'SMITHS.DB8A.CNTL.LOB2DB.KAV2TS'
  UNIT SYSDA;
TSODELETE 'SMITHS.DB2X.UNLD.LOB2DB.KAV2TS';
TEMPLATE UTLREC DSN 'SMITHS.DB8A.UNLD.LOB2DB.KAV2TS'
  UNIT SYSDA;
ADMIN LOBTEMPLATE ADBL1 DSN 'SMITHS.&SSID..&DB..&SN..''
UNIT SYSDA;
ADMIN LOBTEMPLATE ADBL2 DSN 'SMITHS.&SSID..&DB..&SN..'
  UNIT SYSDA;
UNLOAD TABLESPACE LOB2DB.KAV2TS
  FROM TABLE
"SMITHS"."LOB2TB"
  PUNCHDDN(UTLPUNCH)
  UNLDDN(UTLREC);
/*
```

Figure 254. Sample JCL job Log

ADBTEP2 makes the following changes (shown in bold) before passing the JCL to DB2 for processing.

1. The ADMIN LOBTEMPLATE is replaced by TEMPLATE.
2. The UNLOAD syntax is modified.

```
TSODELETE 'SMITHS.DB8A.CNTL.LOB2DB.KAV2TS';
TEMPLATE UTLPUNCH DSN 'SMITHS.DB8A.CNTL.LOB2DB.KAV2TS'
  UNIT SYSDA;
TSODELETE 'SMITHS.DB8A.UNLD.LOB2DB.KAV2TS';
TEMPLATE UTLREC DSN 'SMITHS.DB8A.UNLD.LOB2DB.KAV2TS'
UNIT SYSDA;
TEMPLATE ADBL1 DSN 'SMITHS.&SSID..&DB..&SN..'
UNIT SYSDA;
TEMPLATE ADBL2 DSN 'SMITHS.&SSID..&DB..&SN..'
  UNIT SYSDA;

UNLOAD TABLESPACE LOB2DB.KAV2TS
  FROM TABLE
"SMITHS"."LOB2TB"
 (C2REGULAR,
  C3LOBCOL VARCHAR(255) CLOBF ADBL1,
  C4LOBCOL VARCHAR(255) CLOBF ADBL2)
  PUNCHDDN(UTLPUNCH)
  UNLDDN(UTLREC);
```
Figure 255. ADBTEP changes to job

Overview of ADBTEPA

ADBTEPA is used by DB2 Admin functions such as ALT(alter table columns).

ADBTEPA allows user IDs that are not authorized to certain objects to re-create
those objects if they are implicitly dropped.

For example, when the owner of a table performs an alter to the table that requires
dropping and re-creating the table, any views on this table are also dropped. The
table owner might not have the authority to re-create some or all of the views.
ADBTEPA allows the owner to re-create these views.

The ADBTEPA program receives SQL as input from SYSIN (a batch statement list)
and executes it. In many respects, it is similar to ADBTEP2. For example, they both
use a checkpoint table to record progress through the batch statement list.
ADBTEPA and ADBTEP2 can share the same checkpoint table because the table
definition is identical.

The ADBTEPA program is intended for use with the DB2 Admin authorization
switching function.

Once enabled, ADBTEPA is used by some functions, even if you do not request the
function. ADBTEPA always allows you to perform the same tasks using SQL that
you can perform under your own authorization.

Using ADBTEPA is optional; however, ADBTEPA is required when you use DB2
Admin authorization switching.

Prerequisite: You must enable authorization switching on your DB2 subsystem
before you can use ADBTEPA.

Using ADBTEPA

DB2 Admin generates JCL for ADBTEPA when DB2 Admin authorization switching is enabled.

Prerequisite: ADBTEPA is used only if the auth-switching function is enabled.

The JCL can vary slightly. A user can request an authorization switch by specifying a user ID in the authorization switch ID field on the Alter Parameters panel. Specifying <NONE> indicates that no DB2 Admin authorization switching is requested.

The following figure illustrates an example in which DB2 Admin authorization switching has not been requested, but has been enabled on the subsystem.

```
//CREAT80  EXEC PGM=ADBTEPA,DYNAMNBR=100,
//  PARM='/SSID(DSN7),WORKLIST(GO)'
//STEPLIB  DD DISP=SHR,
//           DSN=ADBB10.SADBLINK
//         DD DISP=SHR,DSN=DSN.DSN7.SDSNEXIT
//         DD DISP=SHR,DSN=DSN.DSN7.SDSNLOAD
//SYSTSPRT DD SYSOUT=*
//ADBPRINT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//ADBOPT   DD *
PLAN=ADBTEPA
//*AUTH_SWITCH_USERID=
//SYSIN    DD DISP=SHR,DSN=ROYC.GO.DDL.CONVMERG
//*
```

Figure 256. DB2 Admin authorization switching example – enabled on subsystem

ADBTEPA, unlike ADBTEP2, is executed directly and not from within DSN under IKJEFT01. Consequently, the SSID PARM is required to identify the DB2 subsystem on which to run. Similarly, the plan that ADBTEPA uses must also be supplied using the ADBOPT DDNAME. ADBTEPA uses the RRSAF attachment to access DB2.

The following figure illustrates the case where an authorization switch ID has been requested to ADBAUTHS.

```
//CREAT80  EXEC PGM=ADBTEPA,DYNAMNBR=100,
//  PARM='/SSID(DSN7),WORKLIST(GO)'
//STEPLIB  DD DISP=SHR,
//           DSN=ADBB10.SADBLINK
//         DD DISP=SHR,DSN=DSN.DSN7.SDSNEXIT
//         DD DISP=SHR,DSN=DSN.DSN7.SDSNLOAD
//SYSTSPRT DD SYSOUT=*
//ADBPRINT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//ADBOPT   DD *
PLAN=ADBTEPA
AUTH_SWITCH_USERID=ADBAUTHS
//SYSIN    DD DISP=SHR,DSN=ROYC.GO.DDL.CONVMERG
```

Figure 257. DB2 Admin authorization switching example – authorization switch requested

In this example, the ID requested was specified using the ADBOPT DDNAME.

ADBTEPA requires that only APF-authorized libraries appear in the STEPLIB, unless ADBTEPA is placed in the link list.

When DB2 Admin authorization switching is enabled, the batch statement list includes system-generated comments near the start of input and after some SQL statements. Do not remove or alter these comments.

Restarting ADBTEPA after a failure

ADBTEPA is restartable in the same way as ADBTEP2.

If it fails, you can change, add, or remove the ADBOPT parameter, `AUTH_SWITCH_USERID=`. Using `AUTH_SWITCH_USERID=` implicitly causes checkpoints to be taken after every statement, even across restarts.

Recommendation: Exercise caution in modifying the batch statement list after a failure. To skip the SQL statement that fails, use the Skip-Next line command within option Z.2B, as opposed to updating the checkpoint record or the batch statement list.

Using automated REORG

Certain DB2 statements can become DB2 pending changes, place the object into an advisory-reorg state, and require a REORG utility to materialize the changes. To help automate the REORG, ADBTEP2 initiates an auto-reorg.

Under certain circumstances, DB2 requires templates for UNLDDN or COPYDDN when performing a REORG. Auto-reorg uses default templates for UNLDDN and COPYDDN. The name is:

```
&USERID..ADBREORG.&DB..&SN..&UNIQ
```

and

```
&USERID..ADBCOPY.&DB..&SN..&UNIQ
```

You can override the templates by including at the beginning of the input stream:

```
ADMIN REORG TEMPLATE  ADBREORG <rest of parameters>
ADMIN COPY TEMPLATE ADBCOPY <rest of parameters>
```

The ADMIN REORG and ADMIN COPY keywords are stripped off the statements.

ADBOPT parameters

ADBOPT parameters are specified using the DDNAME ADBOPT.

Place the options one-per line, and always use uppercase.

ADBOPT parameters for ADBTEP2 and ADBTEPA are listed in the following table:

Table 16. ADBOPT parameters for ADBTEP2 and ADBTEPA

Parameter	Default	Usage	ADBTEP2	ADBTEPA
AUTH_SWITCH_USERID=	None	User ID to provide authority to perform SQL operations.	N/A (Return Code 12 is issued)	Optional
PLAN=	None	Plan that ADBTEPA is to use.	N/A (Ignored)	Mandatory

Table 16. ADBOPT parameters for ADBTEP2 and ADBTEPA (continued)

Parameter	Default	Usage	ADBTEP2	ADBTEPA
COMMIT_ALL=	N	Commit/ checkpoint mode: **Y** commits after every statement. **N** commits before and after non-SQL, or COMMIT statements. After setting this option to **Y**, it persists across restarts.	Optional	Optional (Ignored if AUTH_SWITCH_USERID= is specified)
ADB2UTIL=	ADB2UTIL	Allows alternative name for program ADB2UTIL	Optional	N/A (Ignored)

Pausing ADBTEP2 and ADBTEPA

You can use the ADBPAUSE statement to pause the ADBTEP2 and ADBTEPA programs at a certain point.

To restart ADBTEP2 or ADBTEPA after an ADBPAUSE statement, submit the program again with the RESTART(YES) parameter (either explicitly or by default). The program restarts at the statement that immediately follows the ADBPAUSE statement. If you submit the program using the RESTART(NO) parameter, processing starts at the first statement in the batch statement list.

Chapter 17. Running database utilities

You can use the U.x line command to run DB2 Administration Tool V11.2 - utilities.

You can use the U.x line command on several panels to quickly generate utility job streams.

Topics:
- "Using table space utilities"
- "Using table utilities" on page 419
- "Using index utilities" on page 422
- "Using offline utilities" on page 426
- "Running utilities on LISTDEFs" on page 424

Using table space utilities

Use table space utilities to generate JCL for the utilities that can be run against table spaces.

To display the Table Space Utilities panel, use one of the following commands:
- UTL line command on the Tables Spaces panel (ADB21S). This command allows you to generate utilities for a particular table space.
- UTIL primary command on the Tables Spaces panel (ADB21S). This command allows you to generate utilities for all of the table spaces that are displayed.
- UTIL primary command on the Databases panel (ADB21D). This command allows you to generate utilities for all of the table spaces in the databases that are displayed.
- UT line command on the LISTDEF panel (ADB25L). This command allows you to generate utilities for all of the table spaces or index spaces defined in the LISTDEF.

  In the case of LISTDEF, the Table Utilities panel (ADB25LU) is displayed instead of the Table Space Utilities panel (ADB2US).

Use the Table Space Utilities panel to generate JCL for the utilities that can be run against table spaces. When the JCL is generated, DB2 Admin invokes ISPF edit, which lets you change the JCL, submit it, and copy it to another data set. The following figure shows the Table Space Utilities panel after the UTL line command has been issued.

```
ADB2US in ----------------- DB2X Table Space Utilities ----------------- 23:16
Option ===>

 Execute utility on                                     DB2 System: DSN9
   table space DSN08639.T                               DB2 SQL ID: VNDMPM2
                                                               More:    +
  C - Copy full           CI - Copy incremental   C2 - Copytocopy
 CC - Copy concurrent
  E - Mergecopy           EN - Mergecopy newcopy
  K - Check index         KD - Check data         KL - Check LOB
 LC - Load with Cross loader
  M - Modify              NW - Repair Auxwarn     NX - Repair Auxcheckpend
  N - Repair nocopypend   NA - Repair nocheckpend NB - Repair norcvrpend
 NC - Repair catalog      NL - Repair Levelid     NR - Repair noreorgpend
  O - Reorg               OU - Reorg unload only  OO - Online reorg
 OC - Reorg with Inline Copy
  P - Report recovery      Q - Quiesce
  R - Runstats            RT - Runstats table all RR - Runstats report
 RX
  V - Recover             VC - Recover tocopy     VG - Recover to last GDG
 VI - Rebuild index       VR - Recover torba      VL - Recover logonly
 DG - Define GDG for copy data sets              VP - Recover tologpoint
  U - Unload


 SM - Standard Maintenance  C O R
 BP - Change batch job parameters
 TU - Specify Template Usage

Utility control options
  Review/change options  . . . . . NO   (Yes/No)
  Generate work statement list . . NO   (Yes/No)
  Generate template statements . . NO   (Yes/No)
  Generate modify after copy . . . NO   (Yes/No)
```

Figure 258. Table Space Utilities panel (ADB2US) after issuing the UTL line command

Note: The LC option is displayed only in the following situations:

- The table does not contain XML columns
- The panel is displayed for one table space
- The table space contains only one table
- The table space is not an LOB table space
- The target table does not contain GENERATED ALWAYS columns

When you display the Table Space Utilities panel using the UT line command (as opposed to the UT primary command), it contains an additional option, NL, to set the level identifier. The Specify Utilities Options - REPAIR LEVELID panel (ADB2USN) is displayed with option 4 filled in for you. Press Enter to view the generated JCL in an ISPF edit session. If you scroll down, you can see that the generated REPAIR LEVELID utility control statement exists.

The following options help you to control and vary the utility JCL that will be generated:

BP Enables you to change the default JOB card statements and other system parameters.

TU Enables you to select templates to use for utility JCL and work statement list output.

Review/change options
Use this field to use or review and change the current options for the selected utility. When 'No' is specified, the default options is used for the selected utility.

Generate work statement list

Specify Y to request that the utility control statements be added to a work statement list. Specify N to request an executable utility jobstream.

When you specify the CHECK utility, a batch statement list, which is similar to a work statement list, is generated regardless of the value of this field. The batch statement list is required as an input file to the Batch Restart (ADBTEP2) program, which manages the CHECK utility function.

Generate template statements

Use this field to enable or disable the use of templates.

When you specify the CHECK utility, templates are used regardless of the value of this field because the CHECK utility function requires the use of templates. Either the default templates or the templates that you specify are used.

When you specify Yes to enable the use of templates, DB2 Admin does not generate any TSODELETE statements, which would ensure that any existing data sets for the template are deleted first. To ensure that any existing data sets are deleted, consider using one of the following techniques when you define the template:

- Specify the data set name pattern as a GDG (generation data group) where the next data set in the sequence is generated (+1), and change the other common options so that the GDGLIMIT is 1. This setup will cause the data sets in the group to roll off so that only one data set exists at any one time. For example, a data set pattern name might be specified as &db..&ts..&name..ic(+1).
- Change the other common options to specify a DISP option of NEW, DELETE, DELETE for the data set, if appropriate.

Generate modify after copy

Specify Y to request that utility JCL be generated to run the MODIFY utility after a full image copy is generated. Specify N to suppress the generation of a job step to run the MODIFY utility after a full image copy.

When you specify the CHECK utility, a batch statement list, which is similar to a work statement list, is generated regardless of the value of this field. The batch statement list is required as an input file to the Batch Restart (ADBTEP2) program, which manages the CHECK utility function.

Refer to the online help for detailed information about other options available in this panel.

Tip: When you run the COPY utility, the default is that one copy is written to the data set that is described by the SYSCOPY DD statement. If you want more than one copy of the output, you can create and use templates for the utility data sets COPYDDN 1, COPYDDN2, RECOVERYDDN1, and RECOVERYDDN2.

DB2 Admin supports unloading table (spaces) that produce a record length of less than 32K. When a table (space) with LOB objects is unloaded, it is possible that the required record length exceeds 32K. In this case, you must modify the unload job or WSL to specify the utility statements and parameters that allow unloading the table (space).

Editing generated JCL

Use the Edit Generated JCL panel to edit the JCL you have generated.

The following figure shows the type of output DB2 Admin returns when you generate JCL from the Table Space Utilities panel. In the following figure, option C on the Table Space Utilities panel was chosen (the COPY utility with the FULL parameter specified).

```
 --------------------------------------------------------------------------------
 EDIT       ISTJE.SPFTEMP2.CNTL                            Columns 00001 00072
 Command ===>                                              Scroll ===> PAGE
 000007 //*
 000008 //**************************************************************************
 000009 //*
 000010 //* DB2 ADMIN GENERATED JOB TO RUN COPY ON SELECTED TABLESPACES
 000011 //*
 000012 //**********************************************************ADB2USC***
 000013 //*
 000014 //**************************************************************************
 000015 //* STEP COPY: COPY TABLESPACE DSN8D81A.DSN8S81D
 000016 //**********************************************************ADB2USC1**
 000017 //COPY EXEC DSNUPROC,SYSTEM=DB2X,
 000018 //            LIB='SYS1.DSNDB2X.SDSNLOAD',
 000019 //            UID='ISTJE'
 000020 //DSNUPROC.SYSCOPY DD DSN=ISTJE.DB2X.IC.DSN8D81A.DSN8S81D(+1),
 000021 //            DISP=(NEW,CATLG),
 000022 //            SPACE=(8192,(7,5),RLSE),
 000023 //            UNIT=SYSDA
 000024 //DSNUPROC.SYSIN  DD  *
 000025 COPY TABLESPACE DSN8D81A.DSN8S81D DSNUM ALL FULL YES
 000026 /*
 000027 //**************************************************************************
 000028 //* STEP MOD: MODIFY RECOVERY TABLESPACE DSN8D81A.DSN8S81D
 000029 //**************************************************************************
 000030 //MOD EXEC DSNUPROC,SYSTEM=DB2X,
 000031 //            LIB='SYS1.DSNDB2X.SDSNLOAD',
 000032 //            UID='ISTJE'
 000033 //DSNUPROC.SYSIN  DD  *
 000034 MODIFY RECOVERY TABLESPACE DSN8D81A.DSN8S81D DSNUM ALL
 000035     DELETE AGE(35)
 000036 /*
 ****** **************************** Bottom of Data ****************************
```

Figure 259. Edit generated JCL panel (COPY utility)

Changing batch job utility parameters

Use the Batch Job Utility Parameters panel to change batch job utility parameters.

When you choose option BP on the Table Space Utilities panel, the Batch Job Utility Parameters panel is displayed, as shown in the following figure.

```
ADB2UPA n -------------- DSNA Batch Job Utility Parameters -------------- 12:10
Command ===>


Generate Job Card  . . . YES  (Yes/No)                    DB2 System: DSNA
 Job cards:                                               DB2 SQL ID: J148286
   ===> //J148286D JOB (ACCTINFO,ICE,ICE,ICE),'DB2 UTILITY',CLASS=B,
   ===> //    MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=ACCTINFO,TIME=(,30),
   ===> //    REGION=0M
   ===>
   ===>
 Generate Job CLASS  . . NO   (Yes/No)     JOB CLASS . . . . . . .

JOBPARM:
   ===> S=SY4A
   ===>
   ===>
   ===>


CM Batch EXEC statement parameters:
   Add SSID parameter . . YES     (Yes/No)
   Add PLAN parameter . . YES     (Yes/No)
   Additional parameters to add to CM Batch JCL EXEC statement:
   ===>
   ===>
   ===>
ADBTEP2:
   Restart  . . . . . . .          (Yes/No)
   Maxerrors  . . . . . . 88       (-1 to 99)
   BindError  . . . . . . IGNORE   (MAXE, Save or Ignore)
   Log DIAG . . . . . . . YES      (Yes/No)
   AutoCheck  . . . . . . YES      (Yes/No)
   LOAD Summary Report    YES      (Yes/No)
   Auto Rebuild . . . . . YES      (Yes/No)
   Auto Reorg . . . . . . YES      (Yes/No)
   Advisory Auto Rebuild  YES      (Yes/No)
   Advisory Auto Reorg    YES      (Yes/No)
   Auto Reorg/Rebuild
   after STOGROUP change. YES      (Yes/No)
   LOB/XML IC Unload  . . U        (Error, Use base data)
   Missing IC Unload  . . U        (Error, Use base data)
   Spanned  . . . . . . .          (Yes/No)
   DB2 Pending Changes options:
     Check at DROP  . . . NO       (Yes/No)

Space parameters:
   Unit name  . . . . . . SYSALLDA
   Space unit . . . . . . TRK      (BLK, TRK, CYL or 4096-32760)
   Max Primary  . . . . . 65535    (In above units, 99999999 or blank)
                                    In KB: 3145680
   Max DASD . . . . . . . 65535    (In above units. Allocations beyond this
                                    are sent to tape) In KB: 3145680
   Tape Unit  . . . . . . TAPE     (Unit for tape if size is greater
                                    than Max DASD)
Default space allocation if unable to calculate:
   Primary alloc  . . . . 30       (In above units)
   Secondary alloc  . . . 30       (In above units)

Function-specific parameters:
   Unload pct  . . . ===> 0        (0-99 - % increase for converted data set)
```

Figure 260. Batch Job Utility Parameters panel (ADB2UPA)

On the DB2 Batch Job Utility Parameters panel, you can change the job cards, the
JES2 JCL control statement JOBPARM, the CM Batch EXEC statement parameters,
the ADBTEP2 restart and maximum error specification, and the space parameter
values.

The following options are available:

Generate Job Card

Enter the job cards. If you choose to generate a job card, you can also generate the CLASS parameter. If you select a Job CLASS, the last line of the job cards must end with a comma because DB2 Admin adds an additional line to the job card for the job CLASS.

Generate Job CLASS

If you generate the CLASS parameter, you can specify a job CLASS to override the job CLASS that is specified by the installation.

JOBPARM

If JOBPARM is not specified on this panel, DB2 Admin adds a line for the installation-specified JOBPARM.

CM Batch EXEC statement parameters

Customize the following JCL parameters that used to invoke CM batch.

Add SSID parameter

You can specify:

Yes

Adds the SSID parameter to the EXEC statement. Yes is the default value.

No Does not add the SSID parameter to the EXEC statement.

Add PLAN parameter

You can specify:

Yes

Adds the PLAN parameter to the EXEC statement. Yes is the default value.

No Does not add the PLAN parameter to the EXEC statement.

Additional parameters to add to CM Batch JCL EXEC statement

Specify additional parameters by using the syntax: *parameter_name=value*, where *parameter_name* is the name of the parameter and *value* is its value.

Note: The CM Batch JCL procedure must be predefined to accept any additional JCL procedure parameters that a user might specify.

The following examples illustrate how you might customize the CM batch JCL parameters and the resultant JCL EXEC statement.

Example 1: Suppose the following parameters are specified on the Batch Job Utility Parameters panel:

```
Add SSID parameter . . YES (Yes,No)
Add PLAN parameter . . YES (Yes,No)
Additional parameters to add to CM Batch JCL EXEC statement:
    ===>
    ===>
    ===>
```

The following JCL EXEC statement is generated:

```
//GOCCM    EXEC GOCCM,
// SSID=DSNA,
// PLAN=ADB
```

Example 2: Suppose following parameters are specified on the Batch Job Utility Parameters panel:

```
Add SSID parameter . . NO (Yes,No)
Add PLAN parameter . . NO (Yes,No)
Additional parameters to add to CM Batch JCL EXEC statement:
    ===>
    ===>
    ===>
```

The following JCL EXEC statement is generated:

```
JCL EXEC statement used to invoke CM Batch:
//GOCCM    EXEC GOCCM
```

Example 3: Suppose the following parameters are specified on the Batch Job Utility Parameters panel:

```
Add SSID parameter . . NO  (Yes,No)
Add PLAN parameter . . NO  (Yes,No)
Additional parameters to add to CM Batch JCL EXEC statement:
    ===> PROFILE=DSNA
    ===> PROFILE2=ABC
    ===>
```

The following JCL EXEC statement is generated:

```
JCL EXEC statement used to invoke CM Batch:
//GOCCM    EXEC GOCCM
```

ADBTEP2: Restart

If you select this option you can specify `Yes` or `No` to indicate whether the job is restartable. ADBTEP2 is the Batch Restart program, which provides the ability to restart or resume the execution of an input stream of SQL statements at an intermediate point, in the event that any one of the statements should fail. If you specify `No` for ADBTEP2 restart, a RESTART(NO) parameter is generated for each ADBTEP2 job step.

ADBTEP2: Maxerrors

The number of DSN commands that can fail before the batch restart job ADBTEP2 is stopped:

-1 All errors are ignored. The batch job is not stopped for any error.

0 No errors are allowed. The batch job is stopped on the first error. This is the default value.

1-99
   The specified number of errors are ignored. The batch job is stopped on the next DSN command that fails. For example, if you specify 5, the batch job is stopped when the sixth DSN command fails.

Any failing DSN commands that are ignored are skipped and are written to the ADBHOLD table. When the job ends, if any DSN commands have failed, the restart action field in the checkpoint table indicates that there are held records. Depending on the restart option, the held records are reprocessed when the job is restarted.

ADBTEP2: AutoCheck

Certain SQL or utility operations can place an object into check-pending state. If you set the Autocheck (AC) parameter value to YES, ADBTEP2 will track the statements and processes in the following list that can place an object in check-pending. If one is encountered, ADBTEP2 will perform an automatic CHECK DATA to remove the check-pending state. The default value for AC is NO.

The statements that ADBTEP2 tracks are:

```
ALTER TABLE ...  ADD FOREIGN KEY
ALTER TABLE ....  ADD CONSTRAINT
LOAD REPLACE
LOAD ENFORCE(NO)
RECOVER PIT
```

The processes that ADBTEP2 tracks are:

```
COPY utility  -  perform auto-check prior to COPY
CHECK DATA utility — perform auto-check after CHECKEND
A final auto-check at the end of the SYSIN input stream
```

Restriction: DB2 Admin builds the CHECK DATA statement and all CHECK parameters used during auto-check processing. You cannot specify any other parameters.

Auto Rebuild

The Auto Rebuild parameter determines if the Batch Restart Program initiates a REBUILD of an index when an object is in the RPDB. RPDB*, or PSRBD state.

YES

A REBUILD is attempted.

NO A REBUILD is not attempted.

For more information about the rebuild-pending states, see the *DB2 V10 Utilities Guide*.

Tip: To prevent the ADBTEP2 program from scheduling any automatic REBUILDs, you must set *both* the Auto Rebuild and Advisory Auto Rebuild parameters to No.

Auto Reorg

The Auto Reorg parameter determines if the Batch Restart Program initiates a REORG of a table space when an object is in the REORP state.

YES

A REORG is attempted.

NO A REORG is not attempted.

No is the default.

For more information about the reorg-pending states, see the *DB2 V10 Utilities Guide*.

Tip: To prevent the ADBTEP2 program from scheduling any automatic REORGs you must set *both* the Auto Reorg and Advisory Auto Reorg parameters to No.

Auto Reorg/Rebuild after STOGROUP change

The Auto Reorg/Rebuild after STOGROUP change parameter determines if the Batch Restart Program initiates a REORG or REBUILD after ALTER STOGROUP statement is executed for the table space or index.

YES

A REORG or REBUILD is attempted.

NO A REORG or REBUILD is not attempted.

No is the default.

| **Tip:** To prevent the ADBTEP2 program from scheduling any automatic
| REORGs, you must set Auto Reorg, Advisory Auto Reorg and Auto
| Reorg/Rebuild after STOGROUP change parameters all to No. To prevent the
| ADBTEP2 program from scheduling any automatic REBUILDs, you must set
| Auto Rebuild, Advisory Auto Rebuild and Auto Reorg/Rebuild after
| STOGROUP change parameters all to No.

Advisory Auto Rebuild

The Advisory Auto Rebuild parameter determines if the Batch Restart Program initiates a REBUILD of an index when an object is in the ARBDP state.

YES
A REBUILD is attempted.

NO A REBUILD is not attempted.

No is the default.

For more information about the rebuild-pending states, see the *DB2 V10 Utilities Guide*.

Tip: To prevent the ADBTEP2 program from scheduling any automatic REBUILDs, you must set *both* the Auto Rebuild and Advisory Auto Rebuild parameters to No.

Advisory Auto Reorg

The Advisory Auto Reorg parameter determines if the Batch Restart Program initiates a REORG of a table space when an object is in the AREOR,AREO* state.

YES
A REORG is attempted.

NO A REORG is not attempted.

No is the default.

For more information about the rebuild-pending states, see the *DB2 V10 Utilities Guide*.

Tip: To prevent the ADBTEP2 program from scheduling any automatic REORGs you must set *both* the Auto Reorg and Advisory Auto Reorg parameters to No.

Pending Changes options (DB2 Version 10 New Function mode only):

The Check at DROP parameter controls if a check is made to avoid losing any DB2 pending changes as part of the DROP action.

YES
The DROP is not performed if a DB2 pending change exists.

NO
The DROP is performed without checking for pending changes.

Unit name

The default unit name for new data sets that are allocated.

Space unit

The unit in which space is to be allocated. You can specify that space be allocated in blocks, tracks, cylinders, or a given number of kilobytes.

Max Primary
>The maximum amount of primary space that can be allocated for a data set on DASD, as measured in the specified space unit.

Max DASD
>The maximum amount of space that can be allocated for a data set on DASD, as measured in the specified space unit. When DB2 Admin determines that the amount of estimated space that is required for a data set exceeds this value, the data set is allocated to tape.

Tape unit
>A valid tape unit that has been defined at your site.

Primary alloc
>The default size for primary space allocation when DB2 Admin cannot estimate the space requirements for an allocated data set, such as when the RUNSTATS and STOSPACE utilities have not been run.

Secondary alloc
>The default size for secondary space allocation when DB2 Admin cannot estimate the space requirements for an allocated data set, such as when the RUNSTATS and STOSPACE utilities have not been run.

Unload pct
>Shows the percentage increase for the converted unload data set that the ALT/Object Compare function creates above the UNLOAD data set size. DB2 Admin converts data from the UNLOAD step. The newly converted data might require more space than the unload data set. This parameter allows you to increase the size of the converted data set by a percentage above the unloaded data set, helping to avoid out-of-space conditions.

Specifying utility options

When you use any of the utilities panels, you can choose to display and modify the utility options for the task that you are completing.

For example, you can display the Specify Utility Options panels by following any of these steps:

- Use the UTL line command on a table space to display the Table Space Utilities panel (ADB2US). Choose an option from the menu and specify a Y in the **Review/modify options** field. Press Enter to display the Specify Utility Options panel for that task. For example, if you select the P option (Report Recover), the Specify Utility Options - REPORT RECOVERY panel (ADB2USP) is displayed. You can modify any of the options listed.

- Use the UTL line command on a table to display the Table Utilities panel (ADB2UT). Choose an option from the menu and specify a Y in the **Review/modify options** field. Press Enter to display the Specify Utility Options panel for that task. For example, if you select the UL option (Unload using UNLOAD utility), the Specify Utility Options - UNLOAD panel (ADB2USU) is displayed. You can modify any of the options listed.

- Use the UTL line command on an index to display the Index Utilities panel (ADB2UX). Choose an option from the menu and specify a Y in the **Review/modify options** field. Press Enter to display the Specify Utility Options panel for that task. For example, if you select the K option (Check), the Specify Utility Options - CHECK INDEX panel (ADB2UXK) is displayed. You can modify any of the options listed.

Using utility options for XML and LOBs

Some utility options support XML and LOBs.

The following utility options support XML and LOBs:

CHECK DATA

- Option XMLERROR can provide the values REPORT and INVALIDATE on XML column checks.
- Option PUNCH DD is applicable only when SHRLEVEL is specified as CHANGE. For XML table spaces, before running CHECK DATA, PUNCHDD runs CHECK INDEX on the node ID index of each XML column.
- Option LOBERROR provides the values REPORT and INVALIDATE on LOB column checks.
- Option CLONE indicates that CHECK DATA is to check the clone table in the specified table space. Because clone tables cannot have referential constraints, the utility checks only constraints for inconsistencies between the clone table data and the corresponding LOB data. If you do not specify CLONE, CHECK DATA operates only against the base table.

CHECK INDEX
OPTION CLONE

COPY OPTION CLONE

COPYTOCOPY
OPTION CLONE

LISTDEF
LOB and XML types are supported.

REBUILD INDEX
REBUILD INDEX with SHRLEVEL CHANGE is not allowed for XML Indexes.

REORG
For XML table spaces, and base tables with XML columns, you cannot specify the following options in a REORG statement: DISCARD, REBALANCE, and UNLOAD EXTERNAL.

Using table utilities

Use the Tables Utilities panel to use table utilities.

Use the UTL (utilities) line command or UTL primary command on the Tables, Views, and Aliases panel to display the Table Utilities panel, as shown in the following figure.

Use this panel to generate a batch job stream or work statement list to run one of the displayed utilities against the selected table, view or alias. If you choose to generate a job stream, DB2 Admin invokes an ISPF edit session from which you may further change the contents of the generated job, copy the contents to another data set, or submit it for processing.

Note: If the UX option is used, along with **Generate work statement list: Y**, the LOAD card file which DB2 produces has a reference to a ddname but does not include a TEMPLATE name for it. A TEMPLATE statement must be added

manually.

```
DB2 Admin -------------------- DB2X Table Utilities -------------------- 10:07
Option ===>

 Execute utility on                                 DB2 System: DB2X
   table DSN8810.DEPT                               DB2 SQL ID: ISTJE

   UL  - Unload using UNLOAD utility
   UX  - Unload using REORG UNLOAD EXTERNAL
    L  - Load (with input created from U)
   LX  - Load (with input created from UX or UL)
   LO  - Load (stand-alone, force review/modify options)
   LC  - Load with cross loader (force review/modify options)

   BP - Change batch job parameters
   TU - Specify Template Usage

 Utility control options:
   Review/change options  . . . . . . YES  (Yes/No)
   Generate work statement list . . . NO   (Yes/No)
   Generate template statements . . . NO   (Yes/No)
```

Figure 261. Table Utilities panel (ADB2UT)

Note: The LC option is displayed only in the following situations:

- The table does not contain XML columns
- The panel is displayed for one table only, not for multiple tables
- The target table does not contain GENERATED ALWAYS columns

The following options help you to control and vary the output JCL from the utility:

BP Enables you to change the default JOB card statements and other system parameters.

TU Enables you to select templates to use for utility JCL and work statement list output.

Review/change options
Use this field to use or review and change the current options for the selected utility. When 'No' is specified, the default options is used for the selected utility.

Generate work statement list
Specify Y to request that the utility control statements be added to a work statement list. Specify N to request an executable utility jobstream.

Generate template statements
Use this field to enable or disable the use of templates.

When you specify Yes to enable the use of templates, DB2 Admin does not generate any TSODELETE statements, which would ensure that any existing data sets for the template are deleted first. To ensure that any existing data sets are deleted, consider using one of the following techniques when you define the template:

- Specify the data set name pattern as a GDG (generation data group) where the next data set in the sequence is generated (+1), and change the other common options so that the GDGLIMIT is 1. This setup will cause the data sets in the group to roll off so that only one data set exists at any one time. For example, a data set pattern name might be specified as &db..&ts..&name..ic(+1).

- Change the other common options to specify a DISP option of NEW, DELETE, DELETE for the data set, if appropriate.

Using the **LO** option allows you to create your own LOAD utility job stream. When you select the **LO** option and press Enter, the Specify Utility Options - LOAD panel (ADB2UTC) is displayed, as shown in the following figure.

```
ADB2UTC n ------------- DSNB Specify Utility Options - LOAD ------------- 18:36
Command ===>

Execute utility on table ELACZ.TBTEST1
  using the following options:
                                                               More:     +
Utility ID . . . . .
Unloaded data  . . .
Unloaded how?  . . .          (U - Unload Utility, R - Reorg Utility)
Table/Col info . . .
PRESORTED  . . . . . NO       (Yes/No)
PARALLEL . . . . . .          (Yes, 0-32767)
RESUME . . . . . . .          (Yes/No)
SHRLEVEL . . . . . .          (N - None, C - Change)
REPLACE  . . . . . .          (Yes/No)
  COPYDDN1 . . . . .          (Primary copy DD name)
  COPYDDN2 . . . . .          (Backup copy DD name)
  RECOVERYDDN1 . . .          (Remote primary copy DD name)
  RECOVERYDDN2 . . .          (Remote backup copy DD name)

TABLE schema   . . .          >
       name . . . . .                  > (ALL or ? for table look up)
SAMPLE . . . . . . .          (Percent to sample during RUNSTATS: 1-100)
COLUMN name  . . . .                  > (ALL or ? for column look up)
COLGROUP name  . . .                  > (? for column look up)
  FREQVAL  . . . . .          (Yes/No)
    COUNT  . . . . .          (1-65535)
    OCCUR  . . . . .          (M - Most, B - Both, L - Least)
  HISTOGRAM  . . . .          (Yes/No)
    NUMQUANTILES . .          (1-100, default 100)
INDEX ALL  . . . . .          (Yes/No)
  HISTOGRAM  . . . .          (Yes/No)
    NUMCOLS  . . . .          (1-64, default 1)
    NUMQUANTILES . .          (1-100, default 100)
REPORT . . . . . . .          (Yes/No)
UPDATE . . . . . . .          (A - All, P - Accesspath, S - Space, N - None)

FLASHCOPY  . . . . .          (Y - Yes, N - No, C - Consistent)
KEEPDICTIONARY . . .          (Yes/No)
REUSE  . . . . . . .          (Yes/No)
LOG  . . . . . . . .          (Yes/No/NOC - NOCopypend)
WORKDDN1 . . . . . .          (DD name for temporary work file 1)
WORKDDN2 . . . . . .          (DD name for temporary work file 2)
SORTKEYS . . . . . .          (Estimated no. of keys for parallel sort or NO)
ENFORCE  . . . . . .          (Yes/No)
ERRDDN . . . . . . .          (DD name for error processing)
DISCARDDN  . . . . .          (DD name for discarded records)
DISCARDS . . . . . .          (0 to 2147483647)
SORTDEVT . . . . . .          (Device type for sort work files)
SORTNUM  . . . . . .          (Number of sort work files)
SORTWK . . . . . . .          (0-4)
RBALRSN_CONVERSION            (N - None, B - Basic, E - Extended)
DECFLOAT ROUNDING  .          (Ceiling, Down, Floor, HalfDown, HalfEven,
                                HalfUp, Up)
IMPLICIT_TZ  . . . .          (+/-hh:mm)
```

Figure 262. Specify Utility Options - LOAD panel (ADB2UTC)

DB2 Admin supports unloading table (spaces) that produce a record length of less than 32K. When a table (space) with LOB objects is unloaded, it is possible that the

required record length exceeds 32K. In this case, you must modify the unload job or WSL to specify the utility statements and parameters that allow unloading the table (space).

Related reading: For more information about the fields on the Specify Utility Options - LOAD panel (ADB2UTC), see the Help panel.

Refer to the online help for detailed information about other options available in this panel.

Editing generated JCL

Use the Edit Generated JCL panel to edit the JCL that you have generated.

You then can use standard ISPF editor commands to manually modify the JCL.

The following figure shows the output that DB2 Admin returns when you generate JCL from the Table Utilities panel. In this example, option UX on the Table Utilities panel was chosen (UNLOAD using REORG UNLOAD EXTERNAL).

```
 -------------------------------------------------------------------------------
 EDIT       ISTJE.SPFTEMP2.CNTL                          Columns 00001 00072
 Command ===>                                              Scroll ===> PAGE
 000016 //* STEP DELETE: DELETE OLD DATASETS
 000017 //************************************************************************
 000018 //DELETE   EXEC PGM=IEFBR14
 000019 //SYSREC    DD DSN=ISTJE.DB2X.UNLD.DEPT,
 000020 //             UNIT=SYSDA,DISP=(MOD,DELETE,DELETE),SPACE=(TRK,1)
 000021 //SYSPUNCH  DD DSN=ISTJE.DB2X.CNTL.DSN8D81A.DSN8S81D,
 000022 //             UNIT=SYSDA,DISP=(MOD,DELETE,DELETE),SPACE=(TRK,1)
 000023 //*
 000024 //************************************************************************
 000025 //* STEP UNLOAD: UNLOAD TABLES
 000026 //************************************************************************
 000027 //UNLOAD       EXEC DSNUPROC,SYSTEM=DB2X,
 000028 //             LIB='SYS1.DSNDB2X.SDSNLOAD',
 000029 //             UID='ISTJE'
 000030 //SYSPUNCH  DD DSN=ISTJE.DB2X.CNTL.DSN8D81A.DSN8S81D,
 000031 //             SPACE=(TRK,(5,5),RLSE),
 000032 //             UNIT=SYSDA,
 000033 //             DISP=(,CATLG,DELETE)
 000034 //SYSREC    DD DSN=ISTJE.DB2X.UNLD.DEPT,
 000035 //             DISP=(,CATLG,DELETE),
 000036 //             DCB=(BLKSIZE=8192),
 000037 //             SPACE=(8192,(5,5),RLSE),
 000038 //             UNIT=SYSDA
 000039 //SYSIN     DD    *
 000040 UNLOAD TABLESPACE DSN8D81A.DSN8S81D
 000041    FROM TABLE
 000042 "DSN8810"."DEPT"
 ****** *************************** Bottom of Data ****************************
```

Figure 263. Edit generated JCL panel—UNLOAD utility (ADB2UE)

Using index utilities

Use the Index Utilities panel to use index utilities.

To display the Index Utilities panel, as shown in the following figure, use one of the following commands:
- UTL line command on the Indexes panel (ADB21X). This command allows you to generate utilities for a particular index.

- UTIL primary command on the Indexes panel (ADB21X). This command allows you to generate utilities for all of the indexes that are displayed.
- UTIL IX primary command on the Databases panel (ADB21D). This command allows you to generate utilities for all of the indexes in the databases that are displayed.
- UT line command on the LISTDEF panel (ADB25L). This command allows you to generate utilities for all of the index spaces defined in the LISTDEF.

Use this panel to generate JCL for the utilities that can be run against indexes. When the JCL is generated, DB2 Admin invokes ISPF edit, which enables you to change the JCL, submit it, and copy it to another data set.

```
ADB2UX in -------------------- DSN9 Index Utilities -------------------- 13:17
Option ===>

 Execute utility on                                   DB2 System: DSN9
    all the selected indexes                          DB2 SQL ID: VNDMPM2

   C - Copy full           C2 - Copytocopy
   K - Check
   N - Repair nocopypend    NA - Repair nocheckpend    NB  -Repair norcvrpend
  NR - Repair norbdpend     NO — Repair noreorgpend
   O - Reorg
   R - Runstats             RR - Runstats report
  RX - Runstats (to invalidate dynamic cache)
   V - Recover              RB - Rebuild
   P - Report recovery
  DG - Define GDG for copy data sets


  CL - Create LISTDEF from objects
  BP - Change batch job parameters
  TU - Specify Template Usage

Utility control options:
  Review/change options  . . . . . . YES  (Yes/No)
  Generate work statement list . . . NO   (Yes/No)
  Generate template statements . . . NO   (Yes/No)
```

Figure 264. Index Utilities panel (ADB2UX)

The following options help you to control and vary the output JCL from the utility:

BP Enables you to change the default JOB card statements and other system parameters.

TU Enables you to select templates to use for utility JCL and work statement list output.

Review/change options
Use this field to use or review and change the current options for the selected utility. When 'No' is specified, the default options is used for the selected utility.

Generate work statement list
Specify Y to request that the utility control statements be added to a work statement list. Specify N to request an executable utility jobstream.

When you specify the CHECK utility, a batch statement list, which is similar to a work statement list, is generated regardless of the value of this field. The batch statement list is required as an input file to the Batch Restart (ADBTEP2) program, which manages the CHECK utility function.

Generate template statements
Use this field to enable or disable the use of templates.

When you specify the CHECK utility, templates are used regardless of the value of this field because the CHECK utility function requires the use of templates. Either the default templates or the templates that you specify are used.

Refer to the online help for detailed information about other options available in this panel.

Tip: When you run the COPY utility, the default is that one copy is written to the data set that is described by the SYSCOPY DD statement. If you want more than one copy of the output, you can create and use templates for the utility data sets COPYDDN 1, COPYDDN2, RECOVERYDDN1, and RECOVERYDDN2.

Editing generated JCL

Use the Edit Generated JCL panel to edit the JCL that you have generated.

The following figure shows the type of output that DB2 Admin returns when you generate JCL from the Index Utilities panel. In this example, option R on the Index Utilities panel was chosen (the RUNSTATS utility).

```
 --------------------------------------------------------------------------------
EDIT       ISTJE.SPFTEMP2.CNTL                          Columns 00001 00072
Command ===>                                            Scroll ===> PAGE
****** ***************************** Top of Data ******************************
==MSG>
==MSG>  DB2 Admin: Edit generated JCL
==MSG>
000001 //ISTJED JOB (ADB,OM3),'DB2 UTILITY',
000002 //*       RESTART=stepname, <== For restart remove * and enter step name
000003 //         REGION=0M,NOTIFY=ISTJE,
000004 //         MSGCLASS=H,
000005 //         CLASS=9
000006 //*
000007 //****************************************************************************
000008 //*
000009 //* DB2 ADMIN GENERATED JOB TO RUN RUNSTATS ON INDEXES
000010 //*
000011 //************************************************************ADB2UXR***
000012 //*
000013 //****************************************************************************
000014 //* STEP RUNSTATS: RUNSTATS ON INDEXES
000015 //****************************************************************************
000016 //RUNSTATS EXEC DSNUPROC,SYSTEM=DB2X,
000017 //              LIB='SYS1.DSNDB2X.SDSNLOAD',
000018 //              UID='ISTJE'
000019 //DSNUPROC.SYSIN  DD  *
000020 RUNSTATS INDEX(
000021 "DSN8810"."XDEPT1"
000022          )
****** ***************************** Bottom of Data ***************************
```

Figure 265. Edit generated JCL panel—RUNSTATS utility (ADB2UE)

Running utilities on LISTDEFs

Instead of running utilities against explicitly specified table spaces or indexes, you might want to run the utilities against a predefined LISTDEF.

About this task

To run utilities on a predefined LISTDEF:

Procedure

1. Select option 5 on the Administration Menu panel to display the Utility generation using LISTDEFs and TEMPLATEs panel.
2. Select option L to display the LISTDEFs panel.
3. Issue the UT line command for the desired LISTDEF to display the LISTDEF Utilities panel, as shown in the following figure.

```
DB2 Admin --------------------- DB2X LISTDEF Utilities ------------------ 10:07
Option ===>

 Execute utility using                                    DB2 System: DB2X
   LISTDEF named SYSADM.DBLT0301                           DB2 SQL ID: ISTJE

   C - Copy full            CI - Copy incremental
  CC - Copy concurrent
   E - Mergecopy            EN - Mergecopy newcopy
   K - Check index
   M - Modify
   O - Reorg               OU - Reorg unload only    OO - Online reorg
  OI - Reorg Index
   P - Report recovery
   Q - Quiesce
  RB - Rebuild Index
   R - Runstats Tablespace  RT - Runstats table all   RR - Runstats report
  RX - Runstats (to invalidate dynamic SQL cache for table spaces)
  RI - Runstats Index      RIR - Runstats index report
 RIX - Runstats (to invalidate dynamic SQL cache for index spaces)
   V - Recover             VR - Recover torba        VL - Recover logonly
   U - Unload                                        VP - Recover tologpoint
  SM - Standard Maintenance  C O R
  DG - Define GDG for copy datasets
  BP - Change batch job parameters
  TU - Specify TEMPLATE usage

 Utility control options:
   Review/change options  . . . . . . NO    (Yes/No)
   Generate work statement list . . . NO    (Yes/No)
   Generate template statements . . . NO    (Yes/No)
   Generate tablespace-only steps . . NO    (Yes/No)
```

Figure 266. Table Utilities panel (ADB25LU)

The following options help you to control and vary the output JCL from the utility:

SM   Enables you to specify the base for a generate and establish a series of utilities.

DG   Enables you to specify a GDG (generation data group) base.

BP   Enables you to change the default JOB card statements and other system parameters.

TU   Enables you to select templates to use for utility JCL and work statement list output.

Review/change options
Use this field to use or review and change the current options for the selected utility. When 'No' is specified, the default options are used for the selected utility.

Generate work statement list
Specify Y to request that the utility control statements be added to a work statement list. Specify N to request an executable utility jobstream.

Generate template statements
Use this field to enable or disable the use of templates.

When you specify the CHECK utility, templates are used regardless of the value of this field because the CHECK utility function requires the use of templates. Either the default templates or the templates that you specify are used.

Generate work statement list
Specify Y to request that the utility control statements be added to a work statement list. Specify N to request an executable utility jobstream.

When you specify the CHECK utility, a batch statement list, which is similar to a work statement list, is generated regardless of the value of this field. The batch statement list is required as an input file to the Batch Restart (ADBTEP2) program, which manages the CHECK utility function.

Restriction: Unload jobs generated for LISTDEFS, where the objects contain LOB columns, will fail.

Refer to the online help for detailed information about other options available in this panel.

When the JCL is generated, DB2 Admin invokes ISPF edit, which enables you to change the JCL, submit it, and copy it to another data set.

Related concepts:

Chapter 12, "Using LISTDEFs and TEMPLATEs," on page 259
LISTDEFs are used to specify multiple target objects either by specifying explicit names or patterns of names using wild cards, and TEMPLATEs allow you to define a data set pattern or mask to be used in place of JCL DD statements for various utilities.

Using offline utilities

Offline utilities include DSN1COMP, DSN1COPY, and DSN1PRNT.

About this task

You can use the DB2 Admin table space utility and index space utility panels to invoke the following offline utilities:
- DSN1COMP
- DSN1COPY
- DSN1PRNT

To use these offline utilities:

Procedure

1. From the Table Spaces panel, use the SP line command to display the Table Space Partitions panel (ADB21SP). From the Indexes panel, use the XP line command to display the Index Partitions panel (ADB21XP).
2. Use the UT line command to display the utilities panels (ADB2US for table spaces and ADB2UX for index spaces).
3. Select option DSN1 and press Enter to display the Offline Utilities Selection panel (ADB2US1).
4. Select one of the following offline utilities to run:

   1P DSN1PRNT – Print the following types of data sets:
   - DB2 VSAM data sets that contain table spaces or index spaces

- Image copy data sets
- Sequential data sets that contain DB2 table spaces or index spaces

1C DSN1COPY – Copy the following types of data sets:
- Copy DB2 VSAM data sets to sequential data sets
- Copy DSN1COPY sequential data sets to DB2 VSAM data sets
- Copy DB2 image copy data sets to DB2 VSAM data sets
- Copy DB2 VSAM data sets to other DB2 VSAM data sets
- Copy DSN1COPY sequential data sets to other sequential data sets

The 1C option requires an output data set, defined by a SYSUT2 DD statement. If you do not specify an output data set, DB2 Admin defaults to DUMMY. If you specify an existing data set (DISP=OLD), provide the name and disposition. For a new data set (DISP=NEW), you must also specify, at a minimum, the space units (either TRK or CYL). You can also provide the primary and secondary space allocations and the unit type.

1M DSN1COMP – Estimate space savings as a result of DB2 data compression in table spaces. This option is not available for index spaces.

5. Press Enter to display the Offline Utilities Parameters panel (ADB2USOF).
6. Enter values for the parameters and press Enter to display an ISPF edit session to edit and run the JCL.

Using DB2 High Performance Unload

You can use DB2 High Performance Unload with the MIG function to unload DB2 objects more efficiently.

About this task

You must have DB2 High Performance Unload installed in order to use it.

Procedure

1. On the Migrate Table Spaces panel (ADB28S), issue the NEXT primary command.
2. On the Migrate Parameters panel (ADB28M), issue the UO primary command. The Change Utilities Options panel will be displayed, as shown below.

```
ADB2UOPS  --------------- Change Utilities Options --------------- 12:29
Option ===>

Select one of the following, then press Enter.

   C - Image copy
  KD - Check data
   M - Modify
   O - Reorg tablespace
  OI - Reorg index
   R - Runstats tablespace
   U - Unload
 HPU - High Performance Unload
   L - Load
```

Figure 267. Change Utilities Options panel (ADB2UOPS)

3. On the Change Utilities Options panel, enter the HPU option. The Specify Utilities Options - HPU panel will be displayed, as shown below.

```
ADBPUSH n -------------- Specify Utility Options - HPU ------------- 12:35
Command ===>

Execute utility on migrate object

  using the following options:

DB2 . . . . . . . . . . . YES        (Yes, No, Force)
LOCK  . . . . . . . . . . NO         (Yes/No)
QUIESCE . . . . . . . . . NO         (Yes/No)
PARALLELISM degree for:
  Max partitions  . . . .            (1-65535)
  DB2 SELECT statements . 4545       (1-65535)
  Multiple table spaces . 222        (1-65535)
SPANNED . . . . . . . . . NO         (Yes/No)
  ENFORCE COLUMN ORDER  .            (Yes/No)
FORMAT  . . . . . . . . . VARIABLE   (1-DELIMITED, 2-DSNTIAUL, 3-EXTERNAL,
                                      4-INTERNAL,  5-VARIABLE)
  DELIMITED
    SEP . . . . . . . . . 'P'        (One byte or X'dd')
    DELIM . . . . . . . . X'FF'      (One byte or X'dd')
    NULL DELIM  . . . . . YES        (Yes/No)
  DSNTIAUL
    STRICT  . . . . . . .            (Yes/No)
    LIKE
      Schema  . . . . . . SYSADM   >          (default is MARLINO)
      Name  . . . . . . . PJTBEMP2           > (? to look up)
  VARIABLE  . . . . . . . END        (End/All)
    LIKE
      Schema  . . . . . . SYSADM   >          (default is MARLINO)
      Name  . . . . . . . PJTB1P             > (? to look up)
```

Figure 268. Specify Utility Options - HPU panel (ADBPUSH)

4. On the Specify Utilities Options - HPU panel (ADBPUSH), specify values for your HPU parameters.

5. Press PF3 to return to the Migrate Parameters panel, where you can continue with the MIG process.

Chapter 18. Invoking DB2 EXPLAIN

You can use DB2 Admin to issue SQL EXPLAIN statements, which gather information about the access path that DB2 chooses to process a query, and to use related functions.

Topics:
- "Using the main EXPLAIN panel"
- "Explaining SQL Statements" on page 430
- "Listing rows from a plan table" on page 431
- "Upgrading a plan table" on page 435
- "Creating a plan table" on page 436
- "Creating an index on a plan table" on page 437
- "Creating a statement table" on page 438
- "Creating a function table" on page 439

Using the main EXPLAIN panel

You can use the main EXPLAIN panel to have DB2 explain SQL statements and to perform many other functions.

To start the DB2 Admin EXPLAIN utility, select option E on the Administration Menu panel. The Explain panel is displayed, as shown in the following figure.

```
ADB2E min -------------------------- Explain -------------------------- 10:05
Option ===>


  E - Explain an SQL statement                        DB2 System: DSNB
  L - List PLAN_TABLE   Q - List SYSQUERY explain info DB2 SQL ID: SYSADM
        Schema . . . . . . . . .          >           (default is SYSADM)
        Plan name  . . . . . . .          >           (optional)
        DBRM/package name  . . .          >           (optional)
        Collection ID  . . . . .                   > (optional)

  CT - Create a table used by EXPLAIN
  CX - Create an index for the table
  UT - Upgrade a table to current DB2 version
  CA - Create an alias for the table

  For the above create and upgrade options:
  Schema  . . . . . . . . . . . SYSADM   >           (default is SYSADM)
  Table . . . . . . . . . . . .      1. PLAN_TABLE
                                     2. DSN_STATEMNT_TABLE
                                     3. DSN_FUNCTION_TABLE
                                     4. DSN_STATEMENT_CACHE_TABLE
                                     5. DSN_QUERYINFO_TABLE
                                     6. DSN_PREDICAT_TABLE
                                     7. DSN_USERQUERY_TABLE
                                     8. DSN_PREDICATE_SELECTIVITY
```

Figure 269. Explain panel (ADB2E)

Use this panel to do the following tasks:

- Enter an SQL statement, have DB2 explain the statement, and view the resulting rows in a plan table (PLAN_TABLE).
- List rows from a plan table and see how DB2 will run SQL statements in application plans or packages that were bound with EXPLAIN(YES).

- Create a plan table (a plan table is needed before you can run EXPLAIN statements).
- Upgrade a plan table to the current version of DB2.
- Create an index on the plan table for the DB2 optimizer. An index is recommended if optimizer hints are being used.
- Create a statement table (DSN_STATEMNT_TABLE) in which DB2 EXPLAIN can store the estimated cost for a statement.
- Create a function table (DSN_FUNCTION_TABLE) in which DB2 EXPLAIN can store information on how DB2 resolves function references.
- List queries held in the SYSQUERY table.
- Upgrade a statement table to the current DB2 version.
- Create a query information table that stores information about converted query text if a query is offloaded to an accelerator server.
- Upgrade a function table to the current DB2 version.
- Create an alias for the DB2 EXPLAIN table. The alias allows a user with SELECT and INSERT privileges to populate DB2 EXPLAIN tables that are created under a different AUTHID.
- Create a DSN_STATEMENT_CACHE_TABLE.
- Issue the DB2 BIND QUERY command on SYSQUERY queries. The BIND QUERY panel supports the EXPLAININPUTSCHEMA() clause, which allows you to copy specified rows from an overpopulated PLAN_TABLE to one that should be used solely for BIND QUERY.
- Upgrade a statement cache table.
- Create an index for any of the explain tables.

Explaining SQL Statements

You can request a DB2 EXPLAIN for an SQL statement and view the resulting rows in a plan table.

About this task

To request a DB2 EXPLAIN for an SQL statement and to view the resulting rows in a plan table:

Procedure

1. Select option E on the Explain panel to display the Explain an SQL Statement panel, as shown in the following figure. Optionally use the **SET CURRENT DEGREE** field to set the current degree of parallelism before running the EXPLAIN plan statement. Valid values are 1 and ANY. If the field is left blank, the current degree is not changed.

```
 ┌──────────────────────────────────────────────────────────────────────────┐
 │ DB2 Admin --------------- DB2X Explain an SQL Statement ---------------- 15:50 │
 │ Command ===>                                                               │
 │                                                                            │
 │ SET CURRENT DEGREE =      ;   (Optional)              DB2 System: DB2X     │
 │ EXPLAIN ALL                                           DB2 SQL ID: ISTJE    │
 │                                                                            │
 │ SET QUERYNO =                                                              │
 │ Query number==>                                                            │
 │ FOR                                                                        │
 │ SQL stmt    ==> SELECT * FROM SYSIBM.SYSTABLES WHERE NAME LIKE 'SYS        │
 │                                                                            │
 │                                                                            │
 │                                                                            │
 │                                                                            │
 │                                                                            │
 │                                                                            │
 │ Press ENTER to execute explain, or enter EDIT on the command line to edit  │
 │ the SQL statement.                                                         │
 └──────────────────────────────────────────────────────────────────────────┘
```

Figure 270. Explain an SQL Statement panel (ADB2EE)

2. Enter a query number and an SQL statement. If you leave the query number blank, DB2 Admin generates a query number for you in the form *YYMMDDSSS*, where YYMMDD is the year/month/day and *SSS* is a sequence number.
3. Press Enter to run the EXPLAIN statement. The resulting row in the plan table is shown on the next panel.
4. Use the I line command to display the EXPLAIN results.

Results

You can use the EDIT primary command to edit your SQL statement. When you are in ISPF edit, use the ISPF edit copy commands to copy SQL statements to or from other data sets.

Listing rows from a plan table

You can display the contents of the Plan Table panel.

To display the contents of the Plan Table panel, select option L on the Explain panel. The List Plan Table panel is displayed, as shown in the following figure. The release level and mode of your DB2 subsystem affect the options that are available to you.

```
DB2 Admin ----------- Rows from ISTJE.PLAN_TABLE ------------------ Row 1 of 8
Command ===>  COL HINT INDEX TABLE                           Scroll ===> PAGE

Line commands:
 I - Interpretation  T - Table  X - Index  P - Plan  M - DBRM  K - Package
 DP - Delete rows for plan  DK - Delete for package  DQ - Delete for query no
 ? - Show all line commands

        Query Q  Collect. Progname Pl M Ac M  I T  Table
S      Number Bl (COLLID) (Packg)  No T Ty Co O No Schema     Table Name
           * *  *        *         * * *  * * *  *          *
-- ----------- -- ------- ------- -- - -- -- - -- -------- ----------------
    960125003 1  ADBLCOLI ADBMAIN  1  0 I  0 N 1  SYSIBM     SYSTABLES
    999999999 1  EEEPACK  E41MAIN  1  0 I  1 N 1  SYSIBM     SYSTABLES
    970923001 1  ADBLCOLI ADBMAIN  1  0 I  1 N 1  SYSIBM     SYSTABLES
    981118002 1  ADBL     ADBMAIN  1  0 I  0 N 1  SYSIBM     SYSTABLES
    981118003 1  ADBL     ADBMAIN  1  0 I  0 N 1  SYSIBM     SYSTABLES
    990421001 1  ADBL     ADBMAIN  1  0 I  0 N 1  SYSIBM     SYSTABLES
    990421002 1  ADBL     ADBMAIN  1  0 I  2 N 1  SYSIBM     SYSTABLES
    990421003 1  ADBL     ADBMAIN  1  0 I  2 N 1  SYSIBM     SYSTABLES
****************************** END OF DB2 DATA ******************************
```

Figure 271. An example List Plan Table panel (ADB2EL)

Important: If the dsn_queryinfo_table table exists a panel that is similar to the one
that is shown in the following figure. The dsn_queryinfo_table contains the Explain
information for accelerated queries.

```
ADB2EL in ----------- Rows from J148286.PLAN_TABLE   ------ Row 1 to 41 of 137
Command ===>                                           Scroll ===> PAGE

Commands: HINT  INDEX  COPY  ACCEL
Line commands:
 I - Interpretation  T - Table  X - Index  P - Plan  M - DBRM  K - Package
 DP - Delete rows for plan  DK - Delete for package  DQ - Delete for query no
 ? - Show all line commands

        Query Q  Collect. Progname Pl M Ac M  I  T  Table
S      Number Bl (COLLID) (Packg)  No T Ty Co O  No Schema     Table Name
           * *  *        *         * * *  * * *  *           *
-- ----------- -- -------- -------- -- - -- -- - --- -------- ----------------->
cq     1133 1  JWR      ADB0     1  0 I  2 N  1   SYSIBM   SYSTABLES
       1227 1  JWR      ADB0     1  0 I  2 N  1   SYSACCEL SYSACCELERATEDTABL
       1338 1  JWR      ADB0     1  0 I  0 N  1   SYSIBM   SYSPACKAGE
       1338 1  JWR      ADB0     1  0 I  2 Y  1   SYSIBM   SYSTABLES
```

Figure 272. An example List Plan Table panel (ADB2EL)

Use this panel to see how DB2 will execute SQL statements from previously
executed EXPLAIN statements and from DB2 BIND commands that specify
EXPLAIN(YES).

You can view the information in this panel in five different formats. To switch
between formats, use the following primary commands:
- COL for package mode, which shows Collection (COLLID) and Progname
  (PACKG)
- HINT for hint mode, which shows Hint ID and Hint Used
- INDEX for index information
- TABLE for table information
- ACCEL for accelerator server info
- COPY to copy displayed rows to another PLAN_TABLE

The following fields are available on this panel:

S Input field where you enter one of the line commands listed on the panel.

QUERY NUMBER
A number that identifies the SQL statement.

Q BL
Query block number. Indicates the position of the query in the statement being explained.

APPLNAME (PLAN) or COLLECT. (COLLID) or HINT ID
Name of the application plan for the row, collection ID for the package, hint ID or blank for a dynamic EXPLAIN statement.

PROGNAME (DBRM) or PROGNAME (PACKG) or HINT USED
DBRM name, package name, or hint used.

PL NO
Plan number. Indicates the order in which the EXPLAIN statement will be executed.

MT Method. Indicates the join method to be used.

AC TY
Access type. Indicates the method by which the table will be accessed. This field contains one of the following types:
A	Query is accelerated
I	Index
I1	One-fetch index scan
N	Index scan when the matching predicate contains the IN keyword
R	Table space scan
M	Multiple index scan
MX	Index scan
MI	Intersection of multiple indexes
MU	Union of multiple indexes
Blank	Not applicable to current row

MCO
Matching columns. Indicates the number of index keys used in an index scan.

IO Index only. Whether only the index is accessed in this step or whether data must also be accessed. This field contains one of the following values:
N	No
Y	Yes

T NO
Table number. Indicates the position of the table in the statement.

TABLE SCHEMA
Schema of the table being accessed.

INDEX OWNER
Owner of the index being accessed.

INDEX SCHEMA
Schema of the index being accessed.

TABLE NAME
Name of the table being accessed.

If you select option ACCEL on the Rows from EXPLAIN tables panel, the following additional columns are displayed on the Rows from EXPLAIN tables panel:

RC Reason code for the accelerated query.

Accel Name
   Name of the accelerator server.

Accel Location
   Location of the accelerator server.

Important: To display the interpretation information for accelerated queries, select option I on the Rows from EXPLAIN tables panel. An interpretation panel similar to the panel in the following figure is displayed. Accelerated queries have an access type of A (accesstype = 'A').

```
 ADBPELI n ------- Interpretation of Row from DSN_QUERYINFO_TABLE --------- 10:35
  Command ===>

                                                                    More:      +
                                                              DB2 System: DSNA
  Data as produced by EXPLAIN:

 _ DECLARE C1 CURSOR FOR SELECT  * FROM RAXESHP.TBOC5I03
   ----------------------------------------------------------------------------
  | Query is marked to be offloaded to an accelerator.                         |
  | Query qualifies for routing to an accelerator.                             |
  |                                                                            |
  |                                                                            |
   ----------------------------------------------------------------------------
  Table schema . . : RAXESHP               Table name . . : TBOC5I03
  Accelerator name : ZGRYPHON             Location name  : DB2EC1
  Query number . . : 2                    Query blk no . : 1
  Application name : DSNTEP3              Program name . : DSNTEP3
  Access type  . . : A                    Version  . . . : 2
  Collid . . . . . :                      Group member . :
  Sectnoi  . . . . :                      Seqno  . . . . :
  Explain date . . : 2013-01-31           Explain time . : 04.24.32.67
  Reason code     : 0                     Service info . :
```

Figure 273. Interpretation of Row from DSN_QUERYINFO_TABLE

In addition to the Explain information, the Interpretation panel states whether the query is marked to be offloaded to an accelerator and whether it is qualified to be routed to an accelerator. If the query is not qualified to be offloaded to an accelerator, the reason is stated in the Interpretation of Row from SYSADM.EXPLAIN table.

Copying PLAN_TABLE contents

You can copy PLAN_TABLE rows from one schema to a PLAN_TABLE of a different schema using the explain panels.

To copy the contents of the Plan Table panel, select option L on the Explain panel.

On panel ADB2EL, select the COPY primary command or CQ line command as shown in the following figure. The CQ line command copies all rows that have the same query number, while the COPY primary command copies all of the rows in the table, or a subset of the rows based on the SARG values chosen.

```
ADB2EL in ----------- Rows from J148286.PLAN_TABLE   ------ Row 1 to 41 of 137
Command ===>                                                Scroll ===> PAGE

Commands: HINT  INDEX  COPY  ACCEL
Line commands:
 I - Interpretation  T - Table  X - Index  P - Plan  M - DBRM  K - Package
 DP - Delete rows for plan  DK - Delete for package  DQ - Delete for query no
 ? - Show all line commands

       Query Q  Collect. Progname Pl M Ac M  I  T  Table
S      Number Bl (COLLID) (Packg) No T Ty Co O  No Schema   Table Name
         * *  *        *       * * * * * *  *        *
-- ---------- -- -------- -------- -- - -- -- - --- -------- ----------------->
cq      1133 1  JWR      ADB0     1  0 I  2  N  1 SYSIBM   SYSTABLES
        1227 1  JWR      ADB0     1  0 I  2  N  1 SYSACCEL SYSACCELERATEDTABL
        1338 1  JWR      ADB0     1  0 I  0  N  1 SYSIBM   SYSPACKAGE
        1338 1  JWR      ADB0     1  0 I  2  Y  1 SYSIBM   SYSTABLES
```

Figure 274. List Plan Table panel (ADB2EL)

After selecting the COPY primary command or CQ line command, panel
ADBPELC is displayed, as shown in the following figure:

```
ADBPELC n ---------------------- DSNA Copy entries ---------------- 15:24
Command ===>

 Copy contents from <schema>.PLAN_TABLE                              >
   To schema  . . . . . . . . .            >
   To table name  . . . . . . .                 > (Default PLAN_TABLE)
   Delete rows prior to copy  .     (A - All, M - Matching, N - None)

 Show this panel prior to each use .    (Yes/No)
```

Figure 275. Copy entries panel (ADBPELC)

Depending on how the value of **Show this panel prior to each use** is set, the
panel is displayed when the first of one or more rows are chosen on panel
ADB2EL.

The **Show this panel** option is intended to simplify copying multiple individual
rows to the same target table without asking for the target information for each
row. The option is reset to blank each time panel ADB2EL is first displayed.

The **Delete rows** action is performed after the panel is shown. If the **Show option** is
changed to NO, the **Delete rows** action is performed one time. If the **Show option** is
set to YES, then the **Delete rows** action is performed each time the panel is
displayed. This might mean that the **Delete rows** should be set to All when the
panel is first displayed, and then set to None for subsequent panels so that the
newly copied row from the first display is retained.

Upgrading a plan table

You can upgrade a plan table to the current version of DB2.

To upgrade a plan table to the current version of DB2, select option U on the
Explain Panel. DB2 Admin issues a series of ALTER TABLE PLAN_TABLE ADD
statements to upgrade the plan table so that it contains the maximum number of
columns supported by the current DB2 version.

There is no panel associated with this function. DB2 Admin responds with a
message that indicates whether the plan table was upgraded successfully.

Creating a plan table

Use the Create a Plan Table panel to create a plan table

About this task

To create a plan table:

Procedure

1. Type CX and 1 at the Table option on the Explain panel.

```
ADB2E min -------------------------- Explain -------------------------- 09:03
Option ===>  CX

  E - Explain an SQL statement                          DB2 System: DBAA
  L - List PLAN_TABLE                                   DB2 SQL ID: VNDEJB
        PLAN_TABLE schema  . . .          >             (default is VNDEJB)
        Plan name  . . . . . . .          >             (optional)
        DBRM/package name  . . .          >             (optional)
        Collection ID  . . . . .                    >   (optional)

  CT - Create a table used by EXPLAIN
  CX - Create an index for the table
  UT - Upgrade a table to current DB2 version
  CA - Create an alias for a table

  For the above create and upgrade options:
  Schema  . . . . . . . . . . . .            >              (default is VNDEJB)
  Table . . . . . . . . . . . . 1  1. PLAN_TABLE
                                   2. DSN_STATEMNT_TABLE
                                   3. DSN_FUNCTION_TABLE
                                   4. DSN_STATEMENT_CACHE_TABLE
                                   5. DSN_QUERYINFO_TABLE
```

Figure 276. Explain panel (ADB2E)

2. The Create a Plan Table panel is displayed.

```
ADB2EC in ---------------- DB2X Create DSN_STATEMENT_CACHE_TABLE -------- 09:28
Command ===>

 CREATE TABLE

 Schema  . . . . VNDEJB   >          (optional, default is VNDEJB)
 Name  . . . . . DSN_STATEMENT_CACHE_TABLE

 IN
 Database . . . .               (optional, default is DSNDB04. ? to lookup)
 Table space  . .               (optional, if blank DB2 implicitly creates a TS.
                                  ? to lookup.)
```

Figure 277. Create a Plan Table panel (ADB2EC)

3. Enter the database and table space names you want to use for the plan table. Both names are optional.
4. Press Enter to create the plan table.

What to do next

Refer to the online help for detailed information about the options available in this panel.

Creating an index on a plan table

You can create an index on a plan table for the DB2 optimizer.

About this task

To create an index on the plan table for the DB2 optimizer:

Procedure

1. Type CX and 1 at the Table option on the Explain panel.

```
ADB2E min ------------------------ Explain ------------------------- 09:03
Option ===>  CX

  E - Explain an SQL statement                        DB2 System: DBAA
  L - List PLAN_TABLE                                 DB2 SQL ID: VNDEJB
        PLAN_TABLE schema  . . .         >            (default is VNDEJB)
        Plan name  . . . . . . .         >            (optional)
        DBRM/package name  . . .         >            (optional)
        Collection ID  . . . . .              >  (optional)

  CT - Create a table used by EXPLAIN
  CX - Create an index for the table
  UT - Upgrade a table to current DB2 version
  CA - Create an alias for a table

  For the above create and upgrade options:
  Schema  . . . . . . . . . . . .            >            (default is VNDEJB)
  Table . . . . . . . . . . . . 1  1. PLAN_TABLE
                                   2. DSN_STATEMNT_TABLE
                                   3. DSN_FUNCTION_TABLE
                                   4. DSN_STATEMENT_CACHE_TABLE
                                   5. DSN_QUERYINFO_TABLE
```

Figure 278. Explain panel (ADB2E)

2. The Create Index panel is displayed, as shown in the following figure.

```
ADB26CX n --------------------- DB2X Create Index --------------------- 09:30
Command ===> _____


CREATE INDEX

Schema . . . . . RIVERAF  >          (default is ULVEMAN)
Name . . . . . . IXFGR              > (? to look up)

ON
Table schema . . RIVERAF  >          (default is ULVEMAN)
Table name . . . TBFGR              > (? to look up)

Partitions . . . 0                  (0 for nonpartitioned INDEX)

Like:
Index schema . . _____ >          (required for Like usage)
Index name . . . _____ > (? to look up)
```

Figure 279. Create Index panel (ADB26CX)

3. Specify an index owner and name, a table owner and name, the number of partitions (up to 4096) that the index should contain, and optionally use the LIKE fields to model the index on another index.
4. Press Enter to display the next Create Index panel (ADB21XAR). Specify columns for the index and, optionally, values for the attribute fields.

5. Use the CONTINUE primary command to proceed to the Create Index – Space panel (ADB21XAS). Optionally specify values for the attributes in the ISPF table.
6. Use the CONTINUE primary command to complete the process of creating the index.

What to do next

Refer to the online help for detailed information about the options available in this panel.

Creating a statement table

DB2 EXPLAIN uses a statement table to store the estimated cost for an SQL statement.

About this task

To create a statement table:

Procedure

1. Type CT and 2 at the Table option on the Explain panel.

```
ADB2E min -------------------------- Explain -------------------------- 09:03
Option ===>  CT

  E - Explain an SQL statement                        DB2 System: DBAA
  L - List PLAN_TABLE                                 DB2 SQL ID: VNDEJB
       PLAN_TABLE schema  . . .         >           (default is VNDEJB)
       Plan name  . . . . . . .         >           (optional)
       DBRM/package name  . . .         >           (optional)
       Collection ID  . . . . .               >  (optional)

  CT - Create a table used by EXPLAIN
  CX - Create an index for the table
  UT - Upgrade a table to current DB2 version
  CA - Create an alias for a table

  For the above create and upgrade options:
  Schema  . . . . . . . . . . . .         >           (default is VNDEJB)
  Table . . . . . . . . . . . . .    1. PLAN_TABLE
                                     2. DSN_STATEMNT_TABLE
                                     3. DSN_FUNCTION_TABLE
                                     4. DSN_STATEMENT_CACHE_TABLE
                                     5. DSN_QUERYINFO_TABLE
```

Figure 280. Explain panel (ADB2E)

2. The Create Statement Table panel is displayed.

```
DB2 Admin --------------- DB2X Create DSN_STATEMENT_TABLE -------------- 01:54
Command ===>

 CREATE TABLE

 Owner      ===> ISTJE            (optional, default is ISTJE)
 Name       ===> DSN_STATEMENT_TABLE

 IN
 Database     ===> ISTJEDB  (optional, default is DSNDB04. ? to lookup)
 Table space ===> ISTJESP  (optional, if blank DB2 implicitly creates a TS.
                             ? to lookup.)
```

Figure 281. Create a Statement Table panel (ADB2EC)

3. Accept the defaults or enter your own values.
4. Press Enter to create the statement table.

Creating a function table

DB2 EXPLAIN uses a function table to store information about how function references were resolved.

About this task

To create a function table:

Procedure

1. Type CT and 3 at the Table option on the Explain panel.

```
ADB2E min -------------------------- Explain -------------------------- 09:03
Option ===>  CT

  E - Explain an SQL statement                        DB2 System: DBAA
  L - List PLAN_TABLE                                 DB2 SQL ID: VNDEJB
       PLAN_TABLE schema . . .        >          (default is VNDEJB)
       Plan name . . . . . . .        >          (optional)
       DBRM/package name . . .        >          (optional)
       Collection ID . . . . .              >  (optional)

 CT - Create a table used by EXPLAIN
 CX - Create an index for the table
 UT - Upgrade a table to current DB2 version
 CA - Create an alias for a table

 For the above create and upgrade options:
 Schema . . . . . . . . . . . .           >          (default is VNDEJB)
  Table . . . . . . . . . . . . 3   1. PLAN_TABLE
                                    2. DSN_STATEMNT_TABLE
                                    3. DSN_FUNCTION_TABLE
                                    4. DSN_STATEMENT_CACHE_TABLE
                                    5. DSN_QUERYINFO_TABLE
```

Figure 282. Explain panel (ADB2E)

2. The Create Statement Table panel is displayed.

```
DB2 Admin ---------------- DB2X Create DSN_FUNCTION_TABLE --------------- 01:54
 Command ===>

  CREATE TABLE

  Owner      ===> ISTJE              (optional, default is ISTJE)
  Name       ===> DSN_FUNCTION_TABLE

  IN
  Database    ===> ISTJEDB  (optional, default is DSNDB04. ? to lookup)
  Table space ===> ISTJESP  (optional, if blank DB2 implicitly creates a TS.
                             ? to lookup.)
                            Note: For DB2 Version, specify a table space with
                            an 8K page size or greater.
```

Figure 283. Create a Function Table panel (ADB2EC)

3. Accept the defaults or enter your own values.
4. Press Enter to create the statement table.

Chapter 19. Administering DB2

You can use DB2 Admin to administer your DB2 systems.

Topics:

System Administration panel

DB2 Admin provides an interface to perform many of the tasks required to administer and maintain your DB2 systems.

The tasks that are supported by DB2 Admin are listed in the System Administration panel, as shown in the following figure.

Each option is associated with one or more DB2 commands. For example, when you use the 2D option, DB2 Admin issues the DB2 - DISPLAY THREAD command.

To display this panel, select option Z on the DB2 Administration Menu panel.

```
ADB2Z min ----------------- DSNB System Administration ------------------ 10:12
Option ===>

                                              DB2 System: DB2X
                                              DB2 SQL ID: JSMITH
                                                      More:      +
 DB2 activity related functions:
   2D - Display threads                 2U - Display/terminate utilities
   2T - Display/manage traces           2R - Display/update resource limits
   2S - Stop DB2                        2G - Display group
   2B - Display/manage batch checkpoint 2Z - Manage system parameters
 Buffer pool functions:
   BD - Display buffer pools            BA - Alter buffer pools
   BH - Display buffer pool hit ratios
 Group buffer pool functions:
    GD - Display group buffer pools      GA - Alter group buffer pools
 DB2 log functions:
   LD - Display archive log parameters  LS - Set archive log parameters
   LA - Archive current log             LI - Display log information
   LZ - Set log checkpoint frequency
 DDF functions:
   DU - Display/update CDB              DF - Display DDF
   DC - Display/cancel distributed thds DL - Display active locations
   DT - Start DDF                       DS - Stop DDF
 Stored procedures and functions options:
   PM - Manage stored procedures        FM - Manage functions
 System Backup and Recovery:
   SB - Backup System                   SR - Recover System
   PT - Set Point in Time
 DB2 Accelerator functions:
   AC - Display/manage accelerators     AT - Display accelerated tables
 Security and Audit:
   AP - Manage audit policies
 DB2 autonomic functions:
   RP - Manage RUNSTATS profiles        TW - Manage time windows
   AA - Display alerts                  AH - Display autostats run history



 Note: Before running a command on this panel, make sure you have sufficient
       privilege to execute the related DB2 command.
```

Figure 284. System Administration panel (ADB2Z)

The following tasks are supported by DB2 Admin. They are listed with their associated option numbers.

2D — Display threads
Select this option to display the current status of DB2 threads.

2U — Display/terminate utilities
Select this option to display the status of utility jobs or to terminate utilities.

2T — Display/manage traces
Select this option to display, start, or stop traces.

2R — Display/update resource limits (RLIMIT)
Select this option to display or stop the resource limit (RLIMIT) facility or to update the RLIMIT tables that are created in the system.

2S — Stop DB2
Select this option to stop the DB2 subsystem.

2G — Display Group
Select this option to display information about the data sharing group to which the DB2 subsystem belongs.

2B — Display/manage batch checkpoint table
Select this option to display and manage the checkpoint table (ADBCHKPT)

that is associated with batch jobs that are running ADBTEP2. You can use ADBTEP2 to restart or resume execution of an input stream of SQL statements at an intermediate point in case one of the statements fails.

2Z — Manage system parameters
Select this option to dynamically manage system parameters.

BD — Display buffer pools
Select this option to display the current status of one or more active or inactive buffer pools.

BA — Alter buffer pools
Select this option to alter the attributes of active or inactive buffer pools.

BH — Display buffer pool hit ratios
Select this option to display the hit ratios for the buffer pools.

GD — Display group buffer pools
Select this option to display the group buffer pools for DB2 data sharing. This field is for a data sharing environment only.

GA — Alter group buffer pools
Select this option to alter the group buffer pools for DB2 data sharing. This field is for a data sharing environment only.

LD — Display archive log parameters
Select this option to display information about the input archive log.

LS — Set archive log parameters
Select this option to set the upper limit for the number of and the deallocation time of tape units for the archive log.

LA — Archive current log
Select this option to archive the current DB2 log.

LI — Display log information
Select this option to display information about the DB2 log.

LZ — Set log checkpoint frequency
Select this option to set the DB2 system checkpoint frequency.

DU — Display/update CDB
Select this option to display or update a table in the communications database (CDB).

DF — Display DDF
Select this option to display the status and configuration of the distributed data facility (DDF).

DC — Display/cancel distributed thds
Select this option to display or cancel processing for threads that originate locally and access remote data, or originate remotely and access local data.

DL — Display active locations
Select this option to display statistics about threads with a distributed relationship, or display conversation information about DB2 system threads that interact with VTAM®.

DT — Start DDF
Select this option to start the distributed data facility (DDF) if it has not already been started.

DS — Stop DDF
Select this option to stop the DDF if it has already been started.

PM — Manage stored procedures
Select this option to manage DB2 stored procedures.

FM — Manage functions
Select this option to manage DB2 user-defined functions.

SB — Backup System
Select this option to back up the DB2 subsystem.

PT — Set point in time
Select this option to specify a particular time to which to recover the DB2 subsystem.

SR — Recover System
Select this option to set up a job that will recover the DB2 subsystem to a point in time.

AC - Display/manage accelerators
Select this option to display or update DB2 accelerators.

AT - Display accelerated tables
Select this option to display the DB2 tables that are considered for query offloading to the accelerators.

AP - Manage audit policies
Select this option to display and manage security audit policies for tables or aliases.

RP - Manage RUNSTATS profiles
Select this option to display and manage RUNSTATS profiles for table objects.

TW - Manage time windows
Select this option to display and manage time windows, when the autonomic collection of statistics is allowed.

AA - Display alerts
Select this option to display alerts generated during the autonomic collection of statistics.

AH - Display autostats run history
Select this option to display runstats history generated during autonomic collection of statistics.

Displaying threads

You can display the current status of DB2 threads.

About this task

To display the current status of DB2 threads:

Procedure

1. Select option 2D on the System Administration panel. The Display Threads panel is displayed, as shown in the following figure.

```
ADB2Z2D n ------------------ Display Threads --------------------- 12:10
Command ===>

 -DISPLAY THREAD(
 Connection name . . . .          (name or *, default is TSO)
 TYPE  . . . . . . . . .          (Active, INActive, Indoubt, Postponed,
                                   PROC, SYStem or *)
 LOCATION  . . . . . . .          (name, name* or *)
 LUWID . . . . . . . . .
 DETAIL  . . . . . . . .          (Yes/No)

 Max KB DB2 output . . . 32       (1-1000)

 Output to . . . . . . . T        (T - Table, B – Browse)
 SCOPE . . . . . . . . .          (L – Local, G - Group)
 LIMIT . . . . . . . . .          (Number of lines of output)
```

Figure 285. Display Threads panel (ADB2Z2D)

2. Enter the appropriate keywords and parameters on the panel.

3. Press Enter. DB2 Admin issues the DB2 -DISPLAY THREAD command.

   The information that DB2 Admin returns to you from the command is in ISPF browse format or in a table display panel, depending on what you specify in the **Output to** field.

   If you choose to display the DB2 threads on a table display panel, the Display/Cancel Threads panel (ADB2Z2D2) is displayed, as shown in the following figure. On this panel, you can cancel DB2 threads.

   Restriction: You cannot cancel a thread that is running under the active user ID. An asterisk in the A column indicates which thread is associated with the active user ID.

```
DB2 Admin ------------- DB2X Display/Cancel Threads ---------- Row 1 to 4 of 4
Command ===>                                                 Scroll ===> PAGE

Line commands:
 CAN  - Cancel thread

Sel  Name      St A   Req ID           Auth ID  Plan     ASID Token
     *         *  *     * *             *        *        *    *
---- -------- -- -- ----- ------------ -------- -------- ---- ------
     TSO       T       966 J351156      J351156  TSTDEV   00D6   328
     TSO       T  *      6 ISTJE        ISTJE    ISTJE01  015D   336
CAN  TSO       T       10 DEPT10        DEPT10   D10100   0102   265
     TSO       T        6 JRTESTER      JRTESTER TEST100  00E1   240
***************************** END OF DB2 DATA ******************************
```

Figure 286. Display Threads panel (ADB2Z2D2) – Cancelling a thread

Displaying or terminating utilities

You can display the status of utility jobs or terminate utilities.

About this task

To display the status of utility jobs or terminate utilities:

Procedure

1. Select option 2U on the System Administration panel. The Display/Terminate Utilities panel is displayed, as shown in the following figure.

```
DB2 Admin ---------- DB2X Display/Terminate Utilities ------- ROW 1 TO 1 OF 1
Command ===>                                                  Scroll ===> PAGE

Line commands:
 TERM - Terminate utility   DIS - Display utility

                                                              Time
 Sel  Userid  UtilID  Utility  Stmt  Phase  Count  Status  Jobname  Started
      *       *       *        *  *  *      *      *       *        *
 ---- ------- ------> -------- ----- -----> ------ ------- ---------- --------->
      ISTJE   ISTJE   RUNSTATS    1 RUNSTA 0       ACTIVE  U          2013-05-14
      R148286 R148286 REBUILDI    1 UNLOAD 0       STOPPED PM88750X   2013-07-10
 ****************************** END OF DB2 DATA ******************************
```

Figure 287. Display or Terminate Utilities panel (ADB2Z2U2)

The following fields are available on this panel:

SELECT
Input field where you enter one of the line commands listed on the panel.

USERID
Userid of the person who is running the utility.

UTILITY ID
Utility identifier.

UTILITY
Name of the utility that is currently running.

STMT
Number of the utility statement being processed.

PHASE
Current® phase of the utility, such as RELOAD.

COUNT
Depending on the utility that is currently running, the number of rows, pages, or page sets being processed.

STATUS
Status of the utility, such as ACTIVE.

JOBNAME
The job name that invoked the utility.

TIME STARTED
The date and time when the utility originally started (YYYY-MM-DD-HH:MM:SS).

2. Issue one of the following line commands:
   * TERM to terminate a utility. When you press Enter, DB2 Admin issues the -TERMINATE UTILITY command.
   * DIS to display the status of a utility. DB2 Admin issues the -DISPLAY UTILITY command.

The information that DB2 Admin returns to you from the commands is in ISPF browse format.

Results

The following figure shows the type of information DB2 Admin returns when you issue the DIS line command from the Display/Terminate Utilities panel.

```
DB2 Admin ------- DB2X Browse DB2 Command Output ---  Line 00000000 Col 001 080
 Command ===>                                              Scroll ===> PAGE

 -DIS UTIL(ISTJE)


******************************** TOP OF DATA ********************************
DSNU105I < DSNUGDIS - USERID = ISTJE
                UTILID = ISTJE
                PROCESSING UTILITY STATEMENT 1
                UTILITY = RUNSTATS
                PHASE = RUNSTATS    COUNT = 0
                STATUS = ACTIVE
DSN9022I < DSNUGCCC '-DIS UTIL' NORMAL COMPLETION
****************************** BOTTOM OF DATA *******************************
```

Figure 288. Display Utilities panel (ADB2DB2O)

Displaying or managing traces

You can display, start, or stop traces.

About this task

To display, start, or stop traces:

Procedure

1. Select option 2T on the System Administration panel. The Display/Manager
   Traces panel is displayed, as shown in the following figure.

```
DB2 Admin ------------- DB2X Display/Manage Traces --------------- Row 1 of 2
 Command ===>                                              Scroll ===> PAGE

 Line commands:
  STA - Start trace  STO - Stop trace  DIS - Display trace details

     T  Trace
 Sel No Type  Trace Classes                                 Dest      Qual
     *  *     *                                             *         *
 --- -- ------ -------------------------------------------- ---------- ----
    01 STAT  01,03,04,05                                    SMF        NO
    02 ACCTG 01                                             SMF        NO
***************************** END OF DB2 DATA ******************************
```

Figure 289. Display/Manage Traces panel (ADB2Z2T2)

The following fields are available on this panel:

SEL
   Input field where you enter one of the line commands listed on the panel.

T NO
   Trace number.

TRACE TYPE
   Trace type.

TRACE CLASSES
   Trace classes active for this trace.

DEST
   Destination for the trace.

QUAL

Whether the trace was further qualified.

2. Issue one of the following line commands:

- DIS to display trace details. When you press Enter, DB2 Admin issues the -DISPLAY TRACE command.
- STA to start the trace. When you press Enter, DB2 Admin issues the -START TRACE command.
- STO to STOP the trace. When you press Enter, DB2 Admin issues the -STOP TRACE command.

The information that DB2 Admin returns to you from the commands is in ISPF browse format.

3. If you issue the STA line command, the trace filter panel ADB2Z2TS appears. On this panel, you can specify filters for trace options. The Trace Functions panel is displayed, as shown in the following figure.

```
ADB2Z2TS  ------------------- V91A Trace Functions  ------------------- 08:04
Command ===>
                                                            More:    +
 -START TRACE(
 Trace type . . . . . . .STAT     (STat, ACctg, AUdit, PErfm or MOnitor)
 CLASS  . . . . . . . . .01,03,04
 DEST . . . . . . . . . .SMF      (SMF, GTF, OPn, OPX and/or SRV)
 SCOPE  . . . . . . . . .          (L — Local, G - Group)
 IFCID  . . . . . . . . .
 BUFSIZE  . . . . . . . .          (8-1024)

 TDATA CORRELATION
   Include cor header . .          (Yes/No)
     Include CPU header .          (Yes/No)
     Include trace hdr  .          (Yes/No)
     Include dist hdr . .          (Yes/No)

 COMMENT  . . . . . . . .
 RMID . . . . . . . . . .

 Specify the filters to include or exclude below:
                              Include            Exclude
 PLAN . . . . . . . . . .              >                        >
 AUTHID . . . . . . . . .              >                        >
 LOCATION . . . . . . . .              >                        >
 PKGLOC . . . . . . . . .              >                        >
 PKGCOL . . . . . . . . .              >                        >
 PKGPROG  . . . . . . . .              >                        >
 USERID . . . . . . . . .              >                        >
 APPNAME  . . . . . . . .              >                        >
 WRKSTN . . . . . . . . .              >                        >
 CONNID . . . . . . . . .              >                        >
 CORRID . . . . . . . . .              >                        >
 ROLE . . . . . . . . . .              >                        >
```

Figure 290. Trace Functions (ADB2Z2TS)

Displaying or updating the owner of Resource Limit (RLIMIT) Tables

You can display or update the owner of the resource limit tables.

About this task

To display or update the owner of the resource limit tables:

Procedure

1. Select option 2R on the System Administration panel. The Resource Limit Tables Owner panel is displayed, as shown in the following figure.

```
ADB2Z2R --------------- DB2X Resource Limit Tables Owner -------------- 01:57
Command ===>

                                                        DB2 System: DB2X

Enter the owner of the resource limit tables:

  Owner ===> SYSIBM
```

Figure 291. Resource Limit Tables Owner panel (ADB2Z2R)

2. Enter the owner of the resource limit tables, and press Enter to display the resource limit tables owned by that owner, as shown in the following figure.

```
 ADB2Z2RD ------ DB2X Display/Update Resource Limit Tables -------- Row 1 of 1
Command ===>                                                 Scroll ===> PAGE

                                                        DB2 System: DB2X
Commands:
 DIS - Display RLIMIT  STO - Stop RLIMIT

Line commands:
 S - Display/update  STA - Start RLIMIT with ID   I - Insert row

Select ID Owner    Name              Columns
       *  *        *                      *
------ -- -------- ----------------- -------
       01 SYSIBM   DSNRLMT01               9
       01 SYSIBM   DSNRLST01              11
****************************** END OF DB2 DATA ******************************
```

Figure 292. Display/Update Resource Limit (RLIMIT) Tables panel (ADB2Z2RD)

The following fields are available on this panel:

SELECT
Input field where you enter one of the line commands listed on the panel.

ID

RLIMIT identifier.

OWNER
Authorization ID of the owner of the RLIMIT table.

NAME
Name of the RLIMIT table.

COLUMNS
Number of columns in the RLIMIT table.

3. Issue one of the following commands:

- DIS primary command. Use this command to display the current status of the resource limit. This command is equivalent to the -DISPLAY RLIMIT DB2 command.

    The following figure shows the RLIMIT status information DB2 Admin returns when you issue the DIS primary command.

```
ADB2DB2O  ------- DB2X Browse DB2 Command Output ---  Line 00000000 Col 001 080
Command ===>                                                 Scroll ===> PAGE

  -STA RLIMIT ID=01


******************************* Top of Data **********************************
DSNT704I  #DSN9- SYSIBM.DSNRLST01 HAS BEEN STARTED FOR THE RESOURCE
LIMIT FACILITY
DSNT704I  #DSN9- SYSIBM.DSNRLMT01 HAS BEEN STARTED FOR THE RESOURCE
LIMIT FACILITY
DSN9022I  #DSN9- DSNTCSTR 'START RLIMIT' NORMAL COMPLETION
 ****************************** Bottom of Data ********************************
```

Figure 293. Display RLIMIT panel (ADB2DB2O)

- STO primary command. Use this command to stop the resource limit. This command is equivalent to the -STOP RLIMIT DB2 command.

  The following figure shows the information DB2 Admin returns when you issue the STO primary command to stop the resource limit facility.

```
ADB2DB2O  ------- DB2X Browse DB2 Command Output ---   Line 00000000 Col 001 080
Command ===>                                                 Scroll ===> PAGE

  -STO RLIMIT ID=01


******************************* Top of Data **********************************
DSNT702I  #DSN9- RESOURCE LIMIT FACILITY HAS BEEN STOPPED. WAS USING
SYSIBM.DSNRLST01
DSNT702I  #DSN9- RESOURCE LIMIT FACILITY HAS BEEN STOPPED. WAS USING
SYSIBM.DSNRLMT01
DSN9022I  #DSN9- DSNTCSTP 'STOP RLIMIT' NORMAL COMPLETION
******************************* Bottom of Data *******************************
```

Figure 294. Stop RLIMIT panel (ADB2DB2O)

- S line command. Use this command to display or update the resource limit status.

  The following figure shows the panel returned when you:
  – Issued the S line command to show the content of the RLIMIT table and
  – Used the primary command PRE ON to show the predictive governor columns, as well

```
ADB2Z2RS ---- DB2X Display/Update Resource Limits ID=01 ---------- Row 1 of 1
Command ===>                                                 Scroll ===> PAGE

                                             DB2 System: DB2X
Line commands:  D - Delete  I - Insert  U - Update
                                             F      Reactive B
                                             u       Govenor i
                                             n       Service n
Select Auth ID Plan      Collection       Package LU Name c      Units d
       *       *         *                *       *       *          * *
------ ------->  -------- ----------------->  ------->  -------> - ----------- -
                                                                   ?
       VNDOKAV           COL1              PACK1   LU1    1          ? N
       VNDOKAV           XCOLL             XPACK   XLU    1          ?
       VNDOKAV           YCOLL             YPACK   YLU    1          ?
       VNDWLB1           WLBCOLLECTION     WLBPACKA WLBLU  1          ?
****************************** END OF DB2 DATA *******************************
```

Figure 295. Display RLIMIT panel (ADB2Z2RS)

- S line command. Use this command to display or update the resource limit status of resource limit table DSNRLMT*xx*.

    The following figure shows the panel returned when you:

    – Issued the S line command to show the column values of DSNRLMT*xx* resource table.

```
 ADB2Z2RM  ---- DB2X Display/Update Resource Limits ID=01 ----- Row 1 to 2 of 2
Command ===>                                               Scroll ===> PAGE

                                                     DB2 System: V91A
Line commands:  D - Delete  I - Insert  U - Update
                                                         F Reactive
                                                         u Governor
                                                         n Service
Select User ID  Appl Name       Wrkstn Name    IP          c Units
       *        *               *              *           *       *
------ ------->  --------------> ------------->  -------------->  - ---------
                                                             ?
       SMITHJR  APPL1           WORKSTATN1        30
       PAUL                                     125.123.123.123 8   10
***************************** END OF DB2 DATA *****************************
```

Figure 296. Display RLIMIT panel (ADB2Z2RM)

- STA line command. Use this command start the resource limit with ID.

    The following figure shows the information DB2 Admin returns when you issue the STA line command to start the resource limit facility with a particular ID.

```
 ADB2DB2O ------- DB2X Browse DB2 Command Output ---  Line 00000000 Col 001 080
 Command ===>                                               Scroll ===> PAGE

 -STA RLIMIT ID=01


******************************* Top of Data **********************************
DSNT704I  DB2X SYSIBM.DSNRLST01 HAS BEEN STARTED FOR THE RESOURCE
LIMIT FACILITY
DSN9022I  DB2X DSNTCSTR 'START RLIMIT' NORMAL COMPLETION
****************************** Bottom of Data ********************************
```

Figure 297. Start RLIMIT panel (ADB2DB2O)

- I line command. Use this command to insert a row into the resource limit table.

    The following figure shows the output when you enter the I line command in front of a row from the RLIMIT table in Figure 295 on page 450. On the Insert RLIMIT panel, as shown in the following figure, you can enter values for a new row in the RLIMIT table.

```
ADB2Z2RU ------------------- DB2X Insert RLIMIT ---------------------- 12:05
Command ===>
                                             DB2 System: V91A
                                         DB2 SQL ID: SYSADM
                                             More:     +
Enter/verify details for auth_id.DSNRLSTxx:
  Auth id . . . . .           >            (blank: all)
  Plan name . . . .                        (blank: all)
  Collection . . . .                     > (blank: all)
  Package . . . . .           >            (blank: all)
  LU name . . . . .           >            (blank: local, PUBLIC: all remote)
  Function . . . . .                       (1 - BIND operations
                                            2 - react gov of dyn SQL by package
                                            3 - disable query I/O parallelism
                                            4 - disable query CP parallelism
                                            5 - disables sysplex parallelism
                                            7 - predict. gov. of dyn SQL by pkg)
                                            A - react gov of status SQL by package
  Service units  . . NULL                 (react. gov. limit: 0-2147483647)
  Bind allowed . . .                       (No, for function 1)
  PG warn limit  . . NULL                 (predic. gov. warning limit serv. units
  PG err limit . . . NULL                 (predic. gov. error limit service units
  PG cat B act . . .                       (Execute, Reject, or Warn)

Press ENTER to Insert RLIMIT, or press PF3 to cancel Insert.
```

Figure 298. Insert RLIMIT panel (ADB2Z2RU)

- I line command. Use this command to insert or update column values for the DSNRLMT*xx* resource limit table.

  The following figure shows the output when you enter the I line command in front of a row from the DSNRLMT*xx* RLIMIT table in panel ADB2ZRM.

```
ADB2Z2RI ------------------- DB2X Insert RLIMIT ---------------------- 12:05
Command ===>
                                             DB2 System: V91A
          DB2 SQL ID: SYSADM
Enter/verify details for auth_id.DSNRLMTxx:
  User id . . . . .           >            (blank: all)
  Application name .                     > (blank: all)
  Workstation name .                     > (blank: all)
  IP address . . . .                     > (blank: all)
  Function . . . . .                       (8 - react gov of dyn SQL by client info
                                            9 - pred gov of dyn SQL by client info
                                            B - react gov of static SQL by client info)
  Service units  . . NULL                 (react. gov. limit: 0-2147483647)
  PG warn limit  . . NULL                 (predic. gov. warning limit serv. units)
  PG err limit . . . NULL                 (predic. gov. error limit service units)
  PG cat B act . . .                       (Execute, Reject, or Warn)

Press ENTER to Insert RLIMIT, or press PF3 to cancel Insert
```

Figure 299. Insert RLIMIT panel (ADB2Z2RI)

The information that DB2 Admin returns to you from the commands is in ISPF browse format.

Stopping DB2

You can stop the DB2 subsystem.

About this task

To stop the DB2 subsystem:

Procedure

1. Select option 2S on the System Administration panel. The Stop DB2 panel is displayed, as shown in the following figure.

```
DB2 Admin ----------------------- DB2X Stop DB2 ----------------------- 16:07
Command ===>

 -STOP DB2

  MODE(
  Stop mode          ===>           (Quiesce or Force, default is quiesce)
)
Note: After using FORCE mode, exit from DB2 Admin without issuing any further
SQL statements.
```

Figure 300. Stop DB2 panel (ADB2Z2S)

2. Enter `Quiesce` or `Force` in the **Stop mode** field.
3. Press Enter to stop DB2. DB2 Admin accomplishes this task by issuing the DB2 -STOP DB2 command.

   The information that DB2 Admin returns to you from the command is in ISPF browse format.

Displaying group information

You can display information about the data sharing group to which a DB2 subsystem belongs.

About this task

To display information about the data sharing group to which a DB2 subsystem belongs:

Procedure

Select option 2G on the System Administration panel. The Display Group panel is displayed, as shown in the following figure.

```
  DB2 Admin ------- DB2X Browse DB2 Command Output ---  Line 00000000 Col 001 080
  Command ===>                                            Scroll ===> PAGE

  -DIS GROUP


  ****************************** Top of Data **********************************
  DB2X100I -DB81 DB2XGCMD
  *** BEGIN DISPLAY OF GROUP(DSNDB26 ) GROUP LEVEL(810)
                                    GROUP ATTACH NAME(DB26)
  --------------------------------------------------------------------
  DB2                                  DB2 SYSTEM   IRLM
  MEMBER   ID  SUBSYS CMDPREF    STATUS   LVL NAME     SUBSYS IRLMPROC
  --------  ---  ----  --------  --------  --- --------  ----  --------
  DB81      1  DB81   -DB81      ACTIVE   810 ZPLEX     IR81  DB81IRLM
  DB82      2  DB82   -DB82      FAILED   810 ZPLEX1    IR82  DB82IRLM
  --------------------------------------------------------------------
  SCA   STRUCTURE SIZE:    4096 KB, STATUS= AC,   SCA IN USE:     2 %
  LOCK1 STRUCTURE SIZE:    4096 KB,
  NUMBER  LOCK ENTRIES:    1048576
  NUMBER  LIST ENTRIES:      13878, LIST ENTRIES  IN USE:        22
  *** END DISPLAY OF GROUP(DSNDB26 )
  DSN9022I -DB81 DB2XGCMD 'DISPLAY GROUP ' NORMAL COMPLETION
  ****************************** Bottom of Data *******************************
```

Figure 301. Display Group (ADB2DB2O)

DB2 Admin generates this panel by issuing the DB2 -DISPLAY GROUP command.

Displaying or managing batch checkpoint tables

The DB2 Admin Batch Restart program, ADBTEP2, provides the ability to restart or resume the execution of an input stream of SQL statements, utilities, and DB2 commands in a batch job at an intermediate point, in the event that any one of the statements in that input stream should fail.

About this task

The information to monitor the execution of the input stream is stored in a DB2 table referred to as the checkpoint table.

The Display or Manage Batch Checkpoint Table panel allows you to display and manage the checkpoint table for batch jobs running ADBTEP2. A row exists in the checkpoint table for each active and abnormally terminated job running ADBTEP2.

To display and manage the checkpoint table for the batch jobs that running ADBTEP2:

Procedure

1. Select option 2B on the System Administration panel. The Manage Batch Checkpoint Table panel is displayed, as shown in the following figure.

```
DB2 Admin ------------ DB2X Manage Batch Job Checkpoint Table ----------- 20:39
Option ===>

Batch Job Checkpoint Table :  ADB.ADBCHKPT              DB2 System: DB2X
                                                        DB2 SQL ID: ISTJE

   1 - Display Checkpoint Records
   2 - Display Checkpoint Table Status

 Enter Checkpoint Table Owner:

   Table Owner ===> ADB

 Enter display selection criteria for option 1:

   Userid      ===>                 (default is '')

   Worklist    ===>                 (default is '')
```

Figure 302. Manage Batch Job Checkpoint Table panel (ADB2Z2B)

2. Select one of the following options and press Enter.

- Option 1, Display Checkpoint Records, gives you the ability to display all checkpoint records. Use option 1 to terminate an active ADBTEP2 job, update or delete a record of an abnormal terminated job, or insert a new checkpoint record.

- Option 2, Display Checkpoint Table Status, displays information about the checkpoint table. Use this option to issue any request against the checkpoint table that is supported by DB2 Admin, such as GRANT or REVOKE.

The following figure shows the rows in the table you have selected.

```
DB2 Admin ---------- DB2X Display Batch Job Checkpoint Table ------ Row 1 of 1
Command ===>

                                                       DB2 System: DB2X
                                                       DB2 SQL ID: ISTJE


Checkpoint Table:  ADB.ADBCHKPT

Line commands:
 D - Delete/Terminate  I - Insert  U - Update  N - Skip-Next
                                         Commit      Restart     Restart
S Userid   Worklist Suffix   Time        Number      Command     Action
  *        *        *        *                       *           *
- -------- -------- -------- ---------------> ----------- ----------- ------
  ISTJE    MYMIGR            2002-07-18-16.06           4 COPY        C
  VNDBRON  RI03              2002-07-10-16.19           2
  VNDOJFK  OBJCMP            2002-06-26-16.54           1
  VNDROTH  AAA               2002-06-26-07.36           1 COPY        C
***************************** END OF DB2 DATA *****************************
```

Figure 303. Display Batch Job Checkpoint Table panel (ADB2Z2B1)

When data is unloaded in one job and is then reloaded in another job, the unload suffix has the following format: U*xxxx*. The corresponding reload is R*xxxx*. An additional suffix might also exist, in the format @*xxxx*. Never attempt to update or modify the @*xxxx* record. Delete this record only if you are abandoning a current run of a work statement list. The @*xxxx* record is deleted by the job using R*xxxx*.

Use the following line commands to change the content of the table:

D   To DELETE the row of an abnormally terminated job or to terminate an active job.

I   To INSERT a new row. Row values can be entered on the next panel displayed.

U   To UPDATE the row of an abnormally terminated job. If the job is executing, the request is rejected. Row values can be changed on the next panel.

N   To instruct ADBTEP2 to skip to the next commit instruction.

3. If you use the I or U line commands, the insert or update a checkpoint record panel (ADB2Z2BU) displays. The schema and sqlid values will be used during a restart for setting the current sqlid and current schema special registers at the point of restart. When you update a checkpoint record that does not have a SCHEMA value (is null), the panel value displayed will be blank . If you do not enter a new value, the SCHEMA value remains null. When you insert a new checkpoint record using the panels, if you do not enter a non-blank value, a null value will be stored.

```
ADB2Z2BU DTEST ---------------- INSERT an Entry ----------------------- 23:07
Command ===>

                                                   DB2 System: DSNA
Checkpoint table :  ADB72PAR.ADBCHKPT             DB2 SQL ID: J148286

Enter/Verify:

  Userid . . . . . . J148286
  Worklist . . . . . T14681
  Suffix . . . . . .
  Jobname  . . . . . J148286
  SQLID  . . . . . . J148286
  SCHEMA . . . . . .              >
  Commit number  . . 2
  Restart cmd  . . . New Record
  Restart Act  . . .
  Timestamp  . . . . 2010-05-10-23.05.45.31781
  Server . . . . . . DSNA
  Path . . . . . . . "SYSIBM","SYSFUN","SYSPROC","J148286"



  Session Timezone . ?
  Explain Mode . . . YES
  Program Cntrl  . . NNNINNN


Press ENTER to INSERT an entry, or press PF3 to cancel INSERT.
```

Figure 304. Insert or update a checkpoint record panel (ADB2Z2BU)

Using IBM DB2 Analytics Accelerator

IBM DB2 Analytics Accelerator is an optional workload-optimized appliance add-on that is integrated with DB2 for z/OS. IBM DB2 Analytics Accelerator maximizes performance for long-running complex queries while reducing processor usage.

IBM DB2 Analytics Accelerator for z/OS combines a high-performance hardware platform with an optimized database query engine. The components work together to support a variety of data analysis and business reporting tasks.

You can use DB2 Admin to customize parameters for use with IBM DB2 Analytics Accelerator.

Managing accelerators

You can use DB2 Admin to add, start, stop, display, and delete accelerators. You can also display and save trace information associated with accelerators.

The following topics describe how to manage accelerators using DB2 Admin.

Testing the accelerator connection

You can check whether a real accelerator is properly connected to the DB2 subsystem.

About this task

To verify that the accelerator is connected to DB2, you can run the CONN or PING commands.

Procedure

1. Select option Z - DB2 system administration on the DB2 Administration panel.
2. Select option AC on the System Administration panel.

   The DB2 Accelerators panel is displayed, as shown in the following figure.

```
 ADBPZAC n --------------- DB2X DB2 Accelerators -------------- Row 1 to 2 of 2
 Command ===>                                              Scroll ===> PAGE

 Commands: DIS  ADD
 Line commands:  STA - Start accelerator  STO - Stop accelerator  T - Tables
  DIS - Display accelerator  L - Location  AT - Accelerated tables  DEL - Delete
  ? - Show all line commands

 Select Accelerator Name Location        Type IP Address      Port
        *                *               *    *               *
 ------ ---------------- ---------------- ---- --------------- -----
        ACC1                             V
        ACC2                             R    9.116.85.193     1277
```

Figure 305. DB2 Accelerators panel

3. On the DB2 Accelerators panel, enter one of the following line commands.

Option	Description
CONN	Tests the DRDA connection between DB2 and accelerator
PING	Tests the IP connection between DB2 and accelerator

   An informational message about your connection status will be displayed.

Adding accelerators

You can add a real accelerator or, for testing purposes, you can add a virtual accelerator.

About this task

Virtual accelerators use the EXPLAIN function offered by DB2 for z/OS. Virtual accelerators cannot process regular queries and cannot return query results. However, because virtual accelerators do not require accelerator hardware, you can use them to determine whether queries can be accelerated, check queries for errors,

and estimate query response times. Virtual accelerators must be started with the ACCESS(EXPLAINONLY) statement, and can accept only queries that contain the EXPLAIN statement.

Requirement: After you add a real or virtual accelerator, you must issue the -START ACCEL command to make the accelerator functional.

Procedure

1. Select option Z - DB2 system administration on the DB2 Administration panel.
2. Select option AC on the System Administration panel.

   The DB2 Accelerators panel is displayed, as shown in the following figure.

```
ADBPZAC n --------------- DB2X DB2 Accelerators -------------- Row 1 to 2 of 2
Command ===>                                             Scroll ===> PAGE

Commands: DIS  ADD
Line commands:  STA - Start accelerator  STO - Stop accelerator  T - Tables
 DIS - Display accelerator  L - Location  AT - Accelerated tables  DEL - Delete
 ? - Show all line commands

Select Accelerator Name Location        Type IP Address      Port
       *                *               *    *               *
------ ---------------- ---------------- ---- -------------- -----
       ACC1             DB2EC1           V
       ACC2             DB2EC2           R    9.116.85.193   1277
```

Figure 306. DB2 Accelerators panel

3. On the DB2 Accelerators panel, enter the ADD primary command. The Add Accelerator panel is displayed, as shown in the following figure.

```
ADBPZACA  -------------------- DB2X Add Accelerator ----------------
Command ===>

Enter accelerator details:

Accelerator name  . .
IP address  . . . . .                  > (IPv4 or IPv6 address)
Port  . . . . . . . .                    (numeric)
Pin . . . . . . . . .                    (numeric)
Location  . . . . . .                    (? to lookup)

Start accelerator . .                    (Yes/No, Default - Yes)

Press ENTER to Add accelerator, or PF3 to cancel Add.
```

Figure 307. Add Accelerator panel (ADBPZACA)

4. To create a virtual accelerator, specify only an accelerator name and press Enter. To create a real accelerator, specify all of the parameters on this panel and press Enter. You can create a virtual accelerator by entering only the accelerator name. Entering the IP address, port, and location creates a real accelerator.

   If the accelerator was added successfully, the accelerator information is added to the SYSACCEL.SYSACCELERATORS table and the following message is displayed:

   ```
   Insert stmt executed
   ```

Starting and stopping accelerators

You start and stop an accelerator by using the Start Accelerator and Stop Accelerator panels.

About this task

Before you can use an accelerator, you must start it. After you are done using an accelerator, you might want to stop it to conserve system resources. You might also want to stop an accelerator to terminate inactive accelerator threads.

Procedure

1. Select option Z - DB2 system administration on the DB2 Administration panel.
2. Select option AC on the System Administration panel.
3. On the DB2 Accelerators panel, enter one of the following line commands.

Option	Description
STA	The DB2 Start Accelerator panel is displayed.
STO	The DB2 Stop Accelerator panel is displayed.

If you enter the STA command, the DB2 Start Accelerator panel is displayed, as shown in the following figure.

```
ADBPZADS  ----------------- DB2X Start Accelerator --------------------- 05:42
Command ===>

-START ACCEL

 Accelerator name . . V1                                  >  (name or *)

 MEMBER . . . . . . .              >  (name, only for data sharing environment)

 SCOPE  . . . . . . .                 (L - Local, G - Group,
                                       only for data sharing environment)

 ACCESS . . . . . . . E               (M - Maint, E - Explainonly, or *)
```

Figure 308. DB2 Start Accelerator panel (ADBPZADS)

4. Specify the accelerator that you want to start or stop in the Accelerator name field. To start or stop all accelerators, enter an asterisk (*).

Displaying accelerators
You can display information about the start and stop modes of the accelerators that are connected to your DB2 data server.

Procedure

1. Select option Z - DB2 system administration on the DB2 Administration panel.
2. Select option AC on the System Administration panel.
   The DB2 Accelerators panel is displayed, as shown in the following figure.

```
ADBPZAC n -------------- DB2X DB2 Accelerators -------------- Row 1 to 2 of 2
Command ===>                                              Scroll ===> PAGE

Commands: DIS  ADD
Line commands:  STA - Start accelerator  STO - Stop accelerator  T - Tables
 DIS - Display accelerator  L - Location  AT - Accelerated tables  DEL - Delete
 ? - Show all line commands

Select Accelerator Name Location          Type IP Address      Port
       *                *                 *    *               *
------ ---------------- ---------------- ---- --------------- -----
       ACC1             DB2EC1            V
       ACC2             DB2EC2            R    9.116.85.193    1277
```

Figure 309. DB2 Accelerators panel

3. On the DB2 Accelerators panel, enter the DIS primary command to display information about all of the accelerators, or enter the DIS line command to display information about a particular accelerator.

   The DB2 Display Accelerator panel is displayed, as shown in the following figure.

```
ADBPZADS  ----------------- DB2 Display Accelerator -------------------- 05:39
Command ===>

-DISPLAY ACCEL

 Accelerator name . . *                                      >  (name or *)

 MEMBER . . . . . . .              >  (name, only for data sharing environment)

 SCOPE  . . . . . . .                 (L - Local, G - Group,
                                       only for data sharing environment)

 DETAIL . . . . . . . YES             (Yes/No)

 LIST . . . . . . . .                 (A - Active, or *)
```

Figure 310. DB2 Display Accelerator panel (ADBPZADS)

   If you entered the DIS line command, the Accelerator name field contains the accelerator that you selected. If you entered the DIS primary command, the Accelerator name field contains an asterisk, indicating all accelerators were selected.

4. Press Enter to view information about the accelerators. The information indicates whether each accelerator is in start or stop mode.

Displaying accelerator status

You can display status details about the accelerators that are connected to your DB2 data server.

Procedure

1. Select option Z - DB2 system administration on the DB2 Administration panel.
2. Select option AC on the System Administration panel.

   The DB2 Accelerators panel is displayed, as shown in the following figure.

```
ADBPZAC n -------------- DB2X DB2 Accelerators ------------- Row 1 to 2 of 2
Command ===>                                             Scroll ===> PAGE

Commands: DIS  ADD
Line commands:  STA - Start accelerator  STO - Stop accelerator  T - Tables
 DIS - Display accelerator  L - Location  AT - Accelerated tables  DEL - Delete
 ? - Show all line commands

Select Accelerator Name Location       Type IP Address      Port
       *                *              *    *               *
------ ---------------- ---------------- ---- --------------- -----
       ACC1             DB2EC1          V
       ACC2             DB2EC2          R    9.116.85.193    1277
```

Figure 311. DB2 Accelerators panel

The DB2 Accelerators panel lists the accelerator servers that have been defined to DB2 and provides their location. From this panel, you can perform various actions, including starting and stopping an accelerator server.

3. Enter the I line command in the Select column to view information about the accelerator status.

   The Status Details panel is displayed, as shown in the following figure.

```
ADBPZACI ------------ DB2X Status Details of an Accelerator  ------------ 07:24
Command ===>

Details for accelerator : ACCEL1

State  . . . . . . . . . : ONLINE
Active trace profile . . . : DEFAULT
Active version . . . . . . : 3.1.0.201209030518
Active Netezza version . . : 6.0.5-0.F-1.P-3.Bld-20208
Auth token timestamp . . . : 2013-11-11T10:33:42.487678Z
Accelerator timestamp  . . : 2013-12-03T15:01:04.201483Z

Accelerator setting values:
  Name : ENABLE_FACTREL_PLANNER                 Value : TRUE
  Name : FACTREL_SIZE_THRESHOLD                 Value : 10000000
  Name : MAX_NUM_CONCURRENT_RUNNING_QUERIES     Value : 100
```

Figure 312. Status Details of an Accelerator (ADBPZACI)

Displaying accelerator tasks

You can retrieve a list of tasks that are running on the accelerator. You can view task details and select or cancel tasks.

Procedure

1. Select option Z - DB2 system administration on the DB2 Administration panel.
2. Select option AC on the System Administration panel.

   The DB2 Accelerators panel is displayed, as shown in the following figure.

```
ADBPZAC n -------------- DB2X DB2 Accelerators -------------- Row 1 to 2 of 2
Command ===>                                               Scroll ===> PAGE

Commands: DIS  ADD
Line commands:  STA - Start accelerator  STO - Stop accelerator  T - Tables
 DIS - Display accelerator  L - Location  AT - Accelerated tables  DEL - Delete
 ? - Show all line commands

Select Accelerator Name Location         Type IP Address      Port
       *               *                *    *               *
------ ---------------- ---------------- ---- --------------- -----
       ACC1            DB2EC1           V
       ACC2            DB2EC2           R    9.116.85.193    1277
```

Figure 313. DB2 Accelerators panel

3. Enter the TASK line command next to the accelerator that you want to retrieve task information for.

   The Accelerator Tasks panel is displayed, as shown in the following figure.

```
ADBPZA2 n -------------- DSNB Accelerator Tasks ------------ Row 1 to 10 of 10
Command ===>                                               Scroll ===> CSR

Accelerator name: ACC1

Sel Id        Type          User     Pct   Age Status
    *         *             *        *     * * -
--- --------> ------------> -------- ---- -----> ------------------------>
    505542   LOAD          SYSADM   100      0 Collecting current task
.
.
.
```

Figure 314. DB2 Accelerator Tasks panel (ADBPZA2)

Canceling accelerator tasks

You can cancel tasks that are running on the accelerator.

Procedure

1. Select option Z - DB2 system administration on the DB2 Administration panel.
2. Select option AC on the System Administration panel.

   The DB2 Accelerators panel is displayed, as shown in the following figure.

```
ADBPZAC n -------------- DB2X DB2 Accelerators -------------- Row 1 to 2 of 2
Command ===>                                               Scroll ===> PAGE

Commands: DIS  ADD
Line commands:  STA - Start accelerator  STO - Stop accelerator  T - Tables
 DIS - Display accelerator  L - Location  AT - Accelerated tables  DEL - Delete
 ? - Show all line commands

Select Accelerator Name Location         Type IP Address      Port
       *               *                *    *               *
------ ---------------- ---------------- ---- --------------- -----
       ACC1            DB2EC1           V
       ACC2            DB2EC2           R    9.116.85.193    1277
```

Figure 315. DB2 Accelerators panel

3. Enter the TASK line command next to the accelerator that you want to cancel a task for.

   The Accelerator Tasks panel is displayed, as shown in the following figure.

```
ADBPZA2 n -------------- DB2X Accelerator Tasks ------------ Row 1 to 10 of 10
Command ===>                                              Scroll ===> CSR

Accelerator name: ACC1

Line commands : CAN - Cancel task

Sel Id          Type             User     Pct    Age Status
    *           *                *        *      *  *
--- -------->  --------------->  -------- ---- ----->  ----------------------->
CAN 505542     LOAD             SYSADM   100      0 Collecting current task
.
.
.
```

Figure 316. DB2 Accelerator Tasks panel (ADBPZA2)

4. Enter the CAN line command next to the task that you want to cancel.

Deleting accelerators

You delete accelerators by using the delete command.

Procedure

1. Select option AC on the System Administration panel.

   The DB2 Accelerators panel is displayed.

2. Type the DEL line command next to the accelerator that you want to delete.
   The Delete accelerator confirmation panel is displayed, as shown in the
   following figure.

```
ADB2CONF  -- DB2X Delete accelerator confirmation ----------------- 17:01

Confirm the deletion of accelerator below.
Accelerator: TEST4




Select a choice
  1.  Continue with deletion
  2.  Cancel




 F1=Help     F2=Split    F3=Exit     F9=Swap    F12=Cancel
```

Figure 317. Delete Accelerator confirmation panel (ADB2CONF)

Type 1 to delete the accelerator or 2 to cancel the deletion. If the deletion was
successful, then the following message is displayed:

Delete stmt executed

Creating trace profiles

You can create a trace profile for an accelerator. In a trace profile, you can edit
trace level details and trace components.

Procedure

1. Select option Z - DB2 system administration on the DB2 Administration panel.
2. Select option AC on the System Administration panel.

   The DB2 Accelerators panel is displayed, as shown in the following figure.

```
ADBPZAC n -------------- DB2X DB2 Accelerators ------------- Row 1 to 2 of 2
Command ===>                                                  Scroll ===> PAGE

Commands: DIS  ADD
Line commands:  STA - Start accelerator STO - Stop accelerator  T - Tables
 DIS - Display accelerator  L - Location  AT - Accelerated tables  DEL - Delete
 ? - Show all line commands

Select Accelerator Name Location        Type IP Address     Port
       *                *               *    *              *
------ ---------------- ---------------- ---- -------------- -----
       ACC1             DB2EC1           V
       ACC2             DB2EC2           R    9.116.85.193   1277
```

Figure 318. DB2 Accelerators panel

3. Enter the TPR line command to display a list of trace profiles that are
   associated with the accelerator.

   The DB2 Accelerator Trace Profile panel is displayed, as shown in the following
   figure.

```
ADBPZTPR  ----------- DB2X Accelerator Trace Profiles -------- Row 1 to 8 of 8
 Command ===>                                                  Scroll ===> PAGE

 Accelerator name : ACC1

 Commands: ADD
 Line commands:
  CO - Components   DESC - Description   DEL - Delete  ACT - Activate

                                 Default      File Force
 Sel  Name                 Active Level       Size Flush
      *                    *      *               * *
 ---- --------------------- ------ --------------- ----- -----
      DEFAULT              YES    WARN            3 NO
      QUERY                NO     WARN          160 NO
      LOAD                 NO     WARN           90 NO
      ARCHIVE              NO     WARN           90 NO
      REPLICATION          NO     WARN           63 NO
      SOFTWARE_UPDATE      NO     WARN           20 NO
      CONTROLLERS          NO     WARN           20 NO
      SYSTEM_CRASH         NO     TRACE        1000 YES
```

Figure 319. DB2 Accelerator Trace Profile

4. Enter the ADD primary command to add a new trace profile.

   The DB2 Add Accelerator Trace Profile panel is displayed, as shown in the
   following figure.

```
ADBPZTPA ------ DB2X Add Accelerator Trace Profile ------------ 07:24
Command ===>

Commands: CONTINUE  COMP

Accelerator name . . ACC1
Profile name . . . . QUERY_NEW
Default level  . . .        (D-DEBUG, DE-DEBUG_EXTENDED, E-ERROR, F-FATAL,
                            I-INFO, O-OFF, T-TRACE or W-WARN)
File size in MB  . .        (1-4096)
Force flush  . . . .        (Yes/No)
Description  . . . .
```

Figure 320. DB2 Add Accelerator Trace Profile

5. Enter the COMP command to define individual tracing components whose
   trace levels deviate from the default trace level.

The DB2 Accelerator Trace Profiles panel is displayed, as shown in the following figure.

```
ADBPZTPC  ----------- DB2X Accelerator Trace Profiles ------------- 16:54
Command ===>                                          Scroll ===> PAGE

Accelerator name . . : ACC1
Trace profile name . : QUERY_NEW

Commands: CONTINUE  CANCEL

Line commands:
 I - Insert  R - Repeat  D - Delete

S   Name                          Level         Operation
    *                             *             *
---- ----------------------------- -------------- ----------
    ?                             ?
```

The following component names are valid: CATALOG, CONTROLLER, DRDA, PROFILING, REPLICATION, SERVICES or TRACE.

The following component values are valid: DEBUG, ERROR, INFO, OFF, TRACE or WARN.

What to do next

You can activate a non-default or user-created trace profile by issuing the ACT command. Activating a trace profile allows you to capture and provide customized details to the IBM Support team.

Important: Using a trace profile other than the DEFAULT profile can lower the performance of the accelerator. Use a non-default profile only when instructed to do so by IBM Support. Re-enable the DEFAULT profile when the support activities are complete.

Displaying trace profiles

You can display the list of trace profiles that are associated with an accelerator. From the list, you can select a profile and modify its trace level details.

Procedure

1. Select option Z - DB2 system administration on the DB2 Administration panel.
2. Select option AC on the System Administration panel.

   The DB2 Accelerators panel is displayed, as shown in the following figure.

```
ADBPZAC n --------------- DB2X DB2 Accelerators -------------- Row 1 to 2 of 2
Command ===>                                          Scroll ===> PAGE

Commands: DIS  ADD
Line commands:  STA - Start accelerator  STO - Stop accelerator  T - Tables
 DIS - Display accelerator  L - Location  AT - Accelerated tables  DEL - Delete
 ? - Show all line commands

Select Accelerator Name Location          Type IP Address      Port
       *                *                 *    *               *
------ ---------------- ----------------  ---- --------------- -----
       ACC1             DB2EC1            V
       ACC2             DB2EC2            R    9.116.85.193    1277
```

Figure 321. DB2 Accelerators panel

3. Enter the TPR line command to display a list of trace profiles associated with the accelerator.

   The DB2 Accelerator Trace Profile panel is displayed, as shown in the following figure.

```
ADBPZTPR  ----------- DB2X Accelerator Trace Profiles -------- Row 1 to 8 of 8
 Command ===>                                                 Scroll ===> PAGE

 Accelerator name : ACC1

 Commands: ADD
 Line commands:
  CO - Components   DESC - Description   DEL - Delete  ACT - Activate

                                  Default        File Force
 Sel  Name                 Active Level          Size Flush
      *                    *      *                 *  *
 ---- ---------------------- ------ -------------- ----- -----
      DEFAULT              YES    WARN              3 NO
      QUERY                NO     WARN            160 NO
      LOAD                 NO     WARN             90 NO
      ARCHIVE              NO     WARN             90 NO
      REPLICATION          NO     WARN             63 NO
      SOFTWARE_UPDATE      NO     WARN             20 NO
      CONTROLLERS          NO     WARN             20 NO
      SYSTEM_CRASH         NO     TRACE          1000 YES
```

Figure 322. DB2 Accelerator Trace Profile

4. Enter the CO line command to display the trace profile components.

   The DB2 Accelerator Trace Profile Components panel is displayed, as shown in the following figure.

```
ADBPZTPC ------ DB2X Accelerator Trace Profile Components --------- 07:24
Command ===>

Accelerator name . . : ACC1
Trace profile name . : QUERY

S    Name                          Level
     *                             *
---- ----------------------------- --------------
     QUERY DETAILS                 DEBUG
     QUERY CONTROLLER              DEBUG
```

Figure 323. DB2 Accelerator Trace Profile Components

5. Enter the DESC line command to view a description of the selected trace profile.

Deleting trace profiles

You can delete trace profiles by using the delete command.

Procedure

1. Select option Z - DB2 system administration on the DB2 Administration panel.
2. Select option AC on the System Administration panel.

   The DB2 Accelerators panel is displayed, as shown in the following figure.

```
ADBPZAC n -------------- DB2X DB2 Accelerators ------------- Row 1 to 2 of 2
Command ===>                                                 Scroll ===> PAGE

Commands: DIS  ADD
Line commands:  STA - Start accelerator  STO - Stop accelerator  T - Tables
 DIS - Display accelerator  L - Location  AT - Accelerated tables  DEL - Delete
 ? - Show all line commands

Select Accelerator Name Location        Type IP Address      Port
       *                *               *    *               *
------ ---------------- ---------------- ---- --------------- -----
       ACC1             DB2EC1           V
       ACC2             DB2EC2           R    9.116.85.193    1277
```

Figure 324. DB2 Accelerators panel

3. Enter the TPR line command to display a list of trace profiles associated with the accelerator.

   The DB2 Accelerator Trace Profile panel is displayed, as shown in the following figure.

```
ADBPZTPR  ----------- DB2X Accelerator Trace Profiles -------- Row 1 to 8 of 8
 Command ===>                                                 Scroll ===> PAGE

 Accelerator name : ACC1

 Commands: ADD
 Line commands:
  CO - Components   DESC - Description   DEL - Delete  ACT - Activate

                               Default      File Force
 Sel  Name               Active Level       Size Flush
      *                   *      *              * *
 ---- ----------------------- ------ --------------- ----- -----
      DEFAULT             YES    WARN           3 NO
      QUERY               NO     WARN         160 NO
      LOAD                NO     WARN          90 NO
      ARCHIVE             NO     WARN          90 NO
      REPLICATION         NO     WARN          63 NO
      SOFTWARE_UPDATE     NO     WARN          20 NO
      CONTROLLERS         NO     WARN          20 NO
      SYSTEM_CRASH        NO     TRACE       1000 YES
```

Figure 325. DB2 Accelerator Trace Profile

4. Enter the DEL line command to delete a trace profile.

Retrieving trace data

You can retrieve trace data about the accelerators that are connected to your DB2 data server. Trace data contains information about system status, database runtime, query execution plans, and catalog dumps, which you can use to diagnose accelerator problems.

Procedure

1. Select option Z - DB2 system administration on the DB2 Administration panel.
2. Select option AC on the System Administration panel.

   The DB2 Accelerators panel is displayed, as shown in the following figure.

```
ADBPZAC n --------------- DB2X DB2 Accelerators -------------- Row 1 to 2 of 2
Command ===>                                                   Scroll ===> PAGE

Commands: DIS   ADD
Line commands:  STA - Start accelerator  STO - Stop accelerator  T - Tables
 DIS - Display accelerator  L - Location  AT - Accelerated tables  DEL - Delete
 ? - Show all line commands

Select Accelerator Name Location        Type IP Address       Port
       *                *               *    *                *
------ ---------------- ---------------- ---- --------------- -----
       ACC1             DB2EC1           V
       ACC2             DB2EC2           R    9.116.85.193    1277
```

Figure 326. DB2 Accelerators panel

3. Enter the TR line command next to the accelerator that you want to retrieve trace data for.

   The Accelerator Trace Details panel is displayed, as shown in the following figure.

```
ADBPZATR ------------ DB2X Accelerator Trace Details ------------ 07:24
Command ===>

Trace details for accelerator : ACC1

Save to output data set . . VNDR01.TEST.TRACE

Specify the type of content to save:
  Traces  . . . . . . . . .     (Yes/No)
  Core dumps  . . . . . . .     (Yes/No)
  Netezza core dumps  . . .     (Yes/No)
  Catalog dumps . . . . . .     (Yes/No)
  Execution plans . . . . .     (Yes/No)
  Log history . . . . . . .     (Yes/No)
```

Figure 327. DB2 Accelerator Trace Details panel (ADBPZATR)

   You can specify the types of trace data to save to the specified sequential output data set.

Updating accelerator credentials

You can update the authentication token associated with the accelerator. You may want to update the authentication token periodically to ensure security.

Procedure

1. Select option Z - DB2 system administration on the DB2 Administration panel.
2. Select option AC on the System Administration panel.

   The DB2 Accelerators panel is displayed, as shown in the following figure.

```
ADBPZAC n -------------- DB2X DB2 Accelerators ------------- Row 1 to 2 of 2
Command ===>                                                 Scroll ===> PAGE

Commands: DIS  ADD
Line commands:  STA - Start accelerator  STO - Stop accelerator  T - Tables
 DIS - Display accelerator  L - Location  AT - Accelerated tables  DEL - Delete
 ? - Show all line commands

Select Accelerator Name Location        Type IP Address      Port
       *                *               *    *               *
------ ---------------- --------------- ---- --------------- -----
       ACC1             DB2EC1          V
       ACC2             DB2EC2          R    9.116.85.193    1277
```

Figure 328. DB2 Accelerators panel

3. Enter the UC line command to update the accelerator credentials.

Managing accelerated tables

An accelerated table is a table that is referenced in an accelerated query. You use DB2 Admin to add, display, load, enable, disable, archive, and delete accelerated tables.

You can also view the status of accelerated tables and control the automatic reloading of accelerated tables and incremental updates to accelerated tables.

Information about the accelerated tables is stored in the pseudo-catalog table, SYSACCEL.SYSACCELERATEDTABLES. Each DB2 connection instance has one pseudo-catalog table.

Adding accelerated tables

You can add an accelerated table by using the Add Accelerated Table panel. You must add a table to the accelerator before you can use the accelerator to query table data. You also need to define the tables that are referenced by the query.

Before you begin

You cannot add a DB2 table to an accelerator if any of the following conditions are true:
* The table is not a base table; that is, the value in the TYPE column of the SYSIBM.SYSTABLES table is not T.
* The table uses a row-level security label; that is, the value in the SECURITY_LABEL column of the SYSIBM.SYSTABLES table is R.
* For DB2 for z/OS version 10: The row-level access control is defined for the table; that is, the value in the CONTROL column of the SYSIBM.SYSTABLES table is R or B.

Additional conditions can prevent tables from being added to an accelerator. For a complete list of restrictions, see SYSPROC.ACCEL_ADD_TABLES

Procedure

Use one of the following methods to add a DB2 table to an accelerator:
* Use the following method if you do not know the name of the table that you want to add.
  1. Select option AT on the System Administration panel. The Display Accelerated Tables panel is displayed.

2. From the Display Accelerated Tables panel, issue the ADD primary command to display the Add Accelerated Table panel, as shown in the following figure:

```
ADBPZATA  ----------------- DB2X Add Accelerated Tables ----------------- 17:08
Command ===>

Enter details of table(s) to be defined on an accelerator:

Accelerator name . .                      (? to look up)
Table schema . . . .          >           (Default is ADMF001)
Table name . . . . .                    > (? to look up, * for all tables)

LOAD . . . . . . . .                      (Y - Yes, to load after ADD.
                                           Load will be skipped if
                                           accelerator is virtual)
ENABLE . . . . . . .                      (Y - Yes, to enable after ADD)

Press ENTER to add accelerated tables, or PF3 to cancel add.




  F1=HELP      F2=SPLIT     F3=END      F4=RETURN    F5=RFIND     F6=RCHANGE
  F7=UP        F8=DOWN      F9=SWAP     F10=LEFT     F11=RIGHT    F12=RETRIEVE
```

Figure 329. Add Accelerated Tables panel (ADBPZATA)

Use this panel to define DB2 tables on an accelerator. Information about selected DB2 tables is inserted into the SYSACCEL.SYSACCELERATEDTABLES table.

3. Use the question mark character (?) in the Accelerator name field and Table name field to search for the accelerator you want to add.

- Use the following method to add a specific table from the System Catalog panel.

   1. Select option 1 on the DB2 Administration Menu panel.
   2. Select option T on the System Catalog panel.
   3. Enter the ADDA line command next to the table you want to add.

   The Add Accelerated Table panel (ADBPZATA) is displayed as shown in Figure 329.

Adding accelerated tables during a compare

You can add accelerated tables during a compare.

Procedure

Select option P.CH on the Administration Menu panel.
The Options for Change Functions panel is displayed, as shown in the following figure:

```
ADB2PCO in              Options for Change Functions        11:26
Command ===>

Recreate accelerated tables . . . . . . .     (Yes/No. Default is Yes.)
 Reload accelerated tables . . . . . . . .      (Yes/No. Default is Yes.)
Add new DB2 tables to accelerator . . . .     (Yes/No. Default is Yes.)
Remove deleted tables from accelerator .      (Yes/No. Default is Yes)
Processing order  . . . . . . . . . . . .     (T - object type, H - DB hierarchy.)
                                               (Default is H.)
Statement validation exit name . . . . .      Name of EXEC used to validate
                                                 statments in WSL Validate
```

Figure 330. Options for Change Functions panel (ADB2PCO)

In the **Add new DB2 tables to accelerator** field, enter Yes.

Displaying accelerated tables

You can display information about the tables that are associated with the IBM DB2
Analytics Accelerator.

Procedure

1. Select option AT on the System Administration panel.

   The Display/Manage Accelerated Tables panel is displayed, as shown in the
   following figure. You can specify filters for the tables you want to display by
   entering selection criteria.

```
ADBPZMAT ----------- DB2X Display/Manage Accelerated Tables ------------ 16:13
Option ===>  1


1 - Display accelerated tables                      DB2 System: DB2X
                                                    DB2 SQL ID: SYSADM

Enter display selection criteria.  Settings: LIKE operator;  Criteria not saved
Name . . . . . . .                           > Created by  . .          >
Schema . . . . . .          >                  Accelerator . .          >
Enabled  . . . . .                             Archived  . . .
Created within . .
Altered within . .
Refreshed within .
```

Figure 331. DB2 Display/Manage Accelerated Tables panel (ADBPZMAT)

2. Select option 1 on the Display/Manage Accelerated Tables panel.

   The Display Accelerated Tables panel is displayed, as shown in the following
   figure:

```
ADBPZAT n ----------- DB2X Display Accelerated Tables -------- Row 1 to 1 of 1
Command ===>                                            Scroll ===> PAGE

Commands: RTS  ADD  LOAD  ENABLE  DISABLE  DET
Line commands:
  I - Interpret  AC - Accelerator  T - Table  RTS - RTS info  L - Load
 AR - Archive  EN - Enable  DI - Disable  DE - Delete  DET - Table details
 ? - Show all line commands


    Table            Table    Server       Remote   Remote
S   Name             Schema   Name    E A  Name     Schema   Refresh Time
    *                *        *       * * *          *        *
--- ----------------> -------- -------- - - -------> -------- --------------->
    TBOC5I03         RAXESHP  V1       Y   TBOC5I03 RAXESHP  2013-08-21-06.28.
```

Figure 332. DB2 Display Accelerated Tables panel

3. From the Display Accelerated Tables panel, issue the I line command to display the interpreted values for a table.

The Interpretation of an Object in SYSACCELERATEDTABLES panel is displayed, as shown in the following figure:

```
ADBPZATI --- DB2X Interpretation of an Object in SYSACCELERATEDTABLES ---- 13:14
Command ===>

Details for accelerated table (label): RAXESHP.TBOC5I03

Table name  . . . : TBOC5I03
Table schema  . . : RAXESHP
Server name . . . : V1
Enabled . . . . . : Yes
Archive status  . : Blank - Not archived
Remote name . . . : TBOC5I03_ID1
Remote schema . . : RAXESHP
Created by  . . . : RAXESHP
Support level . . : 3 - Version of the DB2 accelerator server
Created TS  . . . : 2013-08-21-06.28.00.349477
Altered TS  . . . : 2013-08-21-06.28.00.349477
Refresh Time  . . : 2013-08-21-06.28.00.349477
```

Figure 333. Interpretation of an Object in SYSACCELERATEDTABLES panel

In addition to the information that is displayed in the Display Accelerated Tables panel, the Interpretation panel displays the following information:

- The archive status of the table in the accelerator database
- The support level of the DB2 accelerator server when the data in accelerator was created
- The timestamp that identifies when the accelerated table row was inserted in the SYSACCELERATEDTABLES pseudo-catalog table
- The timestamp that identifies when the accelerated table row was last updated in the SYSACCELERATEDTABLES pseudo-catalog table
- The timestamp that identifies when data in the accelerated table was last refreshed

Loading accelerated tables

You must load a table with data after its definition has been copied to the accelerator.

Procedure

1. Select option AT on the System Administration panel.

The Display/Manage Accelerated Tables panel is displayed, as shown in the following figure. You can specify filters for the tables you want to display by entering selection criteria.

```
ADBPZMAT ----------- DB2X Display/Manage Accelerated Tables ------------ 16:13
Option ===>  1


1 - Display accelerated tables                        DB2 System: DB2X
                                                      DB2 SQL ID: SYSADM

Enter display selection criteria.  Settings: LIKE operator;  Criteria not saved
Name . . . . . . .                            > Created by  . .            >
Schema . . . . . .            >                 Accelerator . .            >
Enabled  . . . . .                              Archived  . . .
Created within . .
Altered within . .
Refreshed within .
```

Figure 334. DB2 Display/Manage Accelerated Tables panel (ADBPZMAT)

2. Select option 1 on the Display/Manage Accelerated Tables panel.

   The Display Accelerated Tables panel is displayed, as shown in the following figure.

```
ADBPZAT n ----------- DB2X Display Accelerated Tables -------- Row 1 to 2 of 2
Command ===>                                              Scroll ===> PAGE

Commands: RTS  ADD  LOAD  ENABLE  DISABLE  DET
Line commands:
  I - Interpret  AC - Accelerator  T - Table  RTS - RTS info  L - Load
 AR - Archive  EN - Enable  DI - Disable  DE - Delete  DET - Table details
 ? - Show all line commands

    Table            Table    Server        Remote   Remote
S   Name             Schema   Name    E A Name       Schema   Refresh Time
    *                *        *       * * *          *        *
--- ----------------> -------- -------- - - -------> -------- --------------->
    T1               SYSADM   REAL1    N   T1-ID_16 SYSADM   0001-01-01-00.00
    TEST12344574547459 SYSADM REAL1    N   TEST1234 SYSADM   0001-01-01-00.00
```

Figure 335. DB2 Accelerated Tables panel (ADBPZAT)

3. From the Display Accelerated Tables panel, issue the L (LOAD) line command to load data to a selected table. The LOAD primary command loads data to all of the selected tables.

   Note: You can load DB2 data to your table in batch to reduce wait time. You can run batch by specifying YES for **Run Accelerator functions in batch** on panel ADB2P2.

Enabling and disabling accelerated tables

You can enable or disable an accelerated table to enable or disable query offloading for that DB2 table.

Procedure

1. Select option AT on the System Administration panel.
2. Select option 1 on the Display/Manage Accelerated Tables panel.

   The Display Accelerated Tables panel is displayed, as shown in the following figure.

```
ADBPZAT n ----------- DB2X Display Accelerated Tables -------- Row 1 to 3 of 3
 Command ===>                                                 Scroll ===> PAGE

 Commands: RTS ADD LOAD ENABLE DISABLE DET
 Line commands:
   I - Interpret  AC - Accelerator  T - Table  RTS - RTS info  L - Load
  AR - Archive  EN - Enable  DI - Disable  DE - Delete  DET - Table details
  ? - Show all line commands

     Table             Table     Server         Remote    Remote
 S   Name              Schema    Name    E A   Name       Schema    Refresh Time
     *                 *         *       * * *  *          *         *
 --- ----------------->--------- -------- - - ------->  -------- --------------->
     T1                SYSADM    REAL1    N    T1-ID_16  SYSADM    0001-01-01-00.00
     T2                SYSADM    VIRTUAL1 N    T2_ID1    SYSADM    2013-09-23-15.14
     TEST12344574547459 SYSADM   REAL1    N    TEST1234  SYSADM    0001-01-01-00.00
 ****************************** END OF DB2 DATA ******************************
```

Figure 336. DB2 Accelerated Tables panel (ADBPZAT)

3. On the Display Accelerated Tables panel, enter the EN command to enable an accelerator or the DI command to disable an accelerator, as shown in the following figure:

```
ADBPZAT n ----------- DB2X Display Accelerated Tables -------- Row 1 to 9 of 9
 Command ===>                                                 Scroll ===> PAGE

 Commands: RTS  ADD  LOAD  ENABLE  DISABLE  DET
 Line commands:
   I - Interpret  AC - Accelerator  T - Table  RTS - RTS info  L - Load
  AR - Archive  EN - Enable  DI - Disable  DE - Delete  DET - Table details
  ? - Show all line commands

     Table             Table     Server         Remote    Remote
 S   Name              Schema    Name    E A   Name       Schema    Refresh Time
     *                 *         *       * * *  *          *         *
 -- ------------------ ------->  ------->  - - ------->  -------> --------------->
     DEPTTS            DSN8500   IDAA222  Y A SVL        IDAA2     2012-04-27-13.31.
     DEPTTS2           DSN8500   IDAA1    Y A SVL        IDAA2     2012-04-27-13.31.
     DEPTTS3           DSN8500   IDAA1    Y A SVL        IDAA2     2012-04-27-13.31.
 EN  ITEM10            SCAD22T1  IDAA1    N   ITEM10     SCAD22T1  2013-06-24-12.16.
     T1                S29635_T  IDAA1    Y   T1         S29635_T  2013-06-18-15.55.
 DI  TBADAX06          SCADAX06  IDAA1    Y   TBADAX06   SCADAX06  2013-06-18-16.17.
     TBADGE01_DEPT     VNDRG     ACC1     Y N TBADGE01   VNDDRG    2013-06-07-12.11.
     TBRED1            VNDREDE   IDAA1    N   SVL        IDAA11    2013-06-03-15.20.
     TBRED2            VNDREDE   ACCELERA Y A REMOTENA   REMOTECR  2013-05-30-14.17.
```

Figure 337. DB2 Display Accelerated Tables panel (ADBPZAT)

The status of the accelerated tables is shown in the E (Enable) column. The letter Y in the Enable column indicates that the table is accelerated; the letter N indicates that the table is not accelerated. For example, after you submit the commands in Figure 337, the table ITEM10 is enabled, and the table TBADAX06 is disabled, as shown in the following figure:

```
ADBPZAT n ----------- DB2X Display Accelerated Tables -------- Row 1 to 3 of 3
Command ===>                                              Scroll ===> PAGE

Commands: RTS  ADD  LOAD  ENABLE  DISABLE  DET
Line commands:
  I - Interpret  AC - Accelerator  T - Table  RTS - RTS info  L - Load
 AR - Archive  EN - Enable  DI - Disable  DE - Delete  DET - Table details
 ? - Show all line commands

    Table             Table    Server        Remote   Remote
S   Name              Schema   Name    E A  Name     Schema   Refresh Time
    *                 *        *       * * *         *        *
-- ------------------ -------> -------> - - -------> -------> ---------------->
   DEPTTS             DSN8500  IDAA222  Y A SVL      IDAA2    2012-04-27-13.31.
   DEPTTS2            DSN8500  IDAA1    Y A SVL      IDAA2    2012-04-27-13.31.
   DEPTTS3            DSN8500  IDAA1    Y A SVL      IDAA2    2012-04-27-13.31.
   ITEM10             SCAD22T1 IDAA1    Y   ITEM10   SCAD22T1 2013-06-24-12.16.
   T1                 S29635_T IDAA1    Y   T1       S29635_T 2013-06-18-15.55.
   TBADAX06           SCADAX06 IDAA1    N   TBADAX06 SCADAX06 2013-06-18-16.17.
   TBADGE01_DEPT      VNDRG    ACC1     Y N TBADGE01 VNDDRG   2013-06-07-12.11.
   TBRED1             VNDREDE  IDAA1    N   SVL      IDAA11   2013-06-03-15.20.
   TBRED2             VNDREDE  ACCELERA Y A REMOTENA REMOTECR 2013-05-30-14.17.
****************************** END OF DB2 DATA ******************************
```

Figure 338. DB2 Display Accelerated Tables panel (ADBPZAT)

Archiving accelerated tables

You can archive a table partition to IBM DB2 Analytics Accelerator so that DB2
stores only active data. Archive data is moved to the accelerator to reduce DB2
storage space.

Procedure

1. Select option AT on the System Administration panel.

   The Display/Manage Accelerated Tables panel is displayed, as shown in the
   following figure. You can specify filters for the tables you want to display by
   entering selection criteria.

```
ADBPZMAT ----------- DB2X Display/Manage Accelerated Tables ------------ 16:13
Option ===>  1


1 - Display accelerated tables                        DB2 System: DB2X
                                                      DB2 SQL ID: SYSADM


Enter display selection criteria.  Settings: LIKE operator;  Criteria not saved
Name . . . . . . .                        > Created by . .          >
Schema . . . . . .          >               Accelerator . .         >
Enabled  . . . . .                          Archived  . . .
Created within . .
Altered within . .
Refreshed within .
```

Figure 339. DB2 Display/Manage Accelerated Tables panel (ADBPZMAT)

2. Select option 1 on the Display/Manage Accelerated Tables panel.

   The Display Accelerated Tables panel is displayed, as shown in the following
   figure:

```
ADBPZAT n ----------- DB2X Display Accelerated Tables -------- Row 1 to 2 of 2
Command ===>                                                Scroll ===> PAGE

Commands: RTS  ADD  LOAD  ENABLE  DISABLE  DET
Line commands:
  I - Interpret  AC - Accelerator  T - Table  R - RTS info  L - Load
 AR - Archive  EN - Enable  DI - Disable  DE - Delete  DET - Table details
 ? - Show all line commands

    Table             Table    Server       Remote   Remote
S   Name              Schema   Name    E A  Name     Schema   Refresh Time
    *                 *        *       * * *         *        *
--- ----------------- -------- -------- - - ------->  -------- --------------->
DET SALES             SCADI901 REAL1    N N SALES_ID SCADI901 2013-08-01-10.38
    TBOC5I09          RAXESHP  V1       N   TBOC5I09 RAXESHP  2013-08-01-16.38
***************************** END OF DB2 DATA *****************************
```

Figure 340. DB2 Accelerated Tables panel

3. From the Display Accelerated Tables panel, issue the I line command to display the interpreted values for the table.

    The Interpretation of an Object in SYSACCELERATEDTABLES panel is displayed, as shown in the following figure:

```
ADBPZATI --- DB2X Interpretation of an Object in SYSACCELERATEDTABLES ---- 13:14
Command ===>

Details for accelerated table (label): RAXESHP.TBOC5I03

 ADBPZATI --- DSNA Interpretation of an Object in SYSACCELERATEDTABLES
 Command ===>

 Details for accelerated table (label): RAXESHP.TBOC5I09

 Table name  . . . : TBOC5I09
 Table schema  . . : RAXESHP
 Server name . . . : ACCEL123456
 Enabled . . . . . : No
 Archive . . . . . : A - Accelerator server contains active and archived data
 Remote name . . . : R1
 Remote schema . . : RAXESHP
 Created by  . . . : RAXESHP
 Support level . . : 3 - Version of the DB2 accelerator server
 Created TS  . . . : 2013-01-24-15.20.22.942774
 Altered TS  . . . : 2013-01-24-15.20.22.942774
 Refresh Time  . . : 2013-01-24-15.20.22.942774
```

Figure 341. Interpretation of an Object in SYSACCELERATEDTABLES panel

In addition to the information that is displayed in the Display Accelerated Tables panel, the Interpretation panel displays the following information:

- The archive status of the table in the accelerator database
- The support level of the DB2 accelerator server when the data in accelerator was created
- The timestamp that identifies when the accelerated table row was inserted in the SYSACCELERATEDTABLES pseudo-catalog table
- The timestamp that identifies when the accelerated table row was last updated in the SYSACCELERATEDTABLES pseudo-catalog table
- The timestamp that identifies when data in the accelerated table was last refreshed

Restoring archived accelerated tables

You can restore an archived table to DB2.

Procedure

1. Select option AT on the System Administration panel.

   The Display/Manage Accelerated Tables panel is displayed, as shown in the following figure. You can specify filters for the tables you want to display by entering selection criteria.

```
ADBPZMAT ----------- DB2X Display/Manage Accelerated Tables ------------ 16:13
Option ===>  1


1 - Display accelerated tables                       DB2 System: DB2X
                                                     DB2 SQL ID: SYSADM


Enter display selection criteria.  Settings: LIKE operator;  Criteria not saved
Name . . . . . . .                             > Created by  . .           >
Schema . . . . . .            >                  Accelerator . .           >
Enabled  . . . . .                               Archived  . . .
Created within . .
Altered within . .
Refreshed within .
```

Figure 342. DB2 Display/Manage Accelerated Tables panel (ADBPZMAT)

2. Select option 1 on the Display/Manage Accelerated Tables panel.

   The Display Accelerated Tables panel is displayed, as shown in the following figure:

```
ADBPZAT n ----------- DB2X Display Accelerated Tables -------- Row 1 to 2 of 2
Command ===>                                               Scroll ===> PAGE

Commands: RTS  ADD  LOAD  ENABLE  DISABLE  DET
Line commands:
  I - Interpret  AC - Accelerator  T - Table  RTS - RTS info  L - Load
 AR - Archive  EN - Enable  DI - Disable  DE - Delete  DET - Table details
 ? - Show all line commands


     Table            Table    Server        Remote   Remote
S    Name             Schema   Name     E A  Name     Schema   Refresh Time
     *                *        *        * * *         *        *
---- ---------------- -------- -------- - - ------->  -------- --------------->
RAR  TBT1             SYSADM   AC2      N A TBT1-UID  SYSADM   2015-05-27-14.50
     TBT2             SYSADM   AC2      N   TBT2-UID  SYSADM   0001-01-01-00.00
****************************** END OF DB2 DATA ******************************
```

Figure 343. DB2 Accelerated Tables panel

3. From the Display Accelerated Tables panel, issue the RAR line command to restore an archived table.

   The Restore Partitions panel is displayed, as shown in the following figure:

```
ADBP1ARC  ------------------ DB2X Restore Partitions ------ Row 1 to 10 of 10
Command ===>                                               Scroll ===> PAGE

Commands: ALL  RESET  RESTORE

Table schema . : SYSADM
Table name . . : TBT1

Line commands: S - Select part  D - Deselect part

Input partition range . . .                   (See help for details)

Sel Part   A Restore Limit Key Value
--- ------ - ------- -------------------------------------------------------->
       1                                       '100000'
       2 Y       Y                             '199999'
       3                                       '299999'
       4 Y       Y                             '399999'
       5                                       '499999'
       6                                       '599999'
       7                                       '699999'
       8                                       '799999'
       9                                       '899999'
      10                                       '999999'
****************************** END OF DB2 DATA ******************************
```

Figure 344. DB2X Restore Partitions panel

4. In the Restore Partitions panel, issue the S line command to select the archived partitions that you want to restore. In column A, a Y value indicates that the table is archived. In the Restore column, a Y value indicates that the table will be restored.

5. After you have selected the archived partitions, issue the RESTORE primary command.

Deleting accelerated tables

You can issue a line command to delete DB2 tables from the accelerator; that is, remove it from the accelerated tables, so that query offloading can be disabled for those tables.

Procedure

1. Select option AT on the System Administration panel.

   The Display/Manage Accelerated Tables panel is displayed, as shown in the following figure. You can specify filters for the tables you want to display by entering selection criteria.

```
ADBPZMAT ----------- DB2X Display/Manage Accelerated Tables ------------ 16:13
Option ===>  1


1 - Display accelerated tables                      DB2 System: DB2X
                                                    DB2 SQL ID: SYSADM

Enter display selection criteria.  Settings: LIKE operator;  Criteria not saved
Name . . . . . . .                            > Created by . .          >
Schema . . . . . .            >                 Accelerator . .          >
Enabled  . . . . .                              Archived  . . .
Created within . .
Altered within . .
Refreshed within .
```

Figure 345. DB2 Display/Manage Accelerated Tables panel (ADBPZMAT)

2. Select option 1 on the Display/Manage Accelerated Tables panel.

The Display Accelerated Tables panel is displayed, as shown in the following figure.

```
ADBPZAT n ----------- DB2X Display Accelerated Tables -------- Row 1 to 2 of 2
 Command ===>                                            Scroll ===> PAGE

 Commands: RTS  ADD  LOAD  ENABLE  DISABLE  DET
 Line commands:
   I - Interpret  AC - Accelerator  T - Table  RTS - RTS info  L - Load
  AR - Archive  EN - Enable  DI - Disable  DE - Delete  DET - Table details
  ? - Show all line commands

    Table              Table    Server        Remote   Remote
 S  Name               Schema   Name    E A Name      Schema   Refresh Time
    *                  *        *       * * *         *        *
--- ------------------ -------- -------- - - -------> -------- --------------->
    SALES              SCADI901 REAL1    N N SALES_ID SCADI901 2013-08-01-10.38
    TBOC5I09           RAXESHP  V1       N   TBOC5I09 RAXESHP  2013-08-01-16.38
***************************** END OF DB2 DATA ******************************
```

Figure 346. Display Accelerated Tables panel (ADBPZAT)

3. From the Display Accelerated Tables panel, issue the DE line command to delete a table. A "Delete successful" message is issued if no errors are detected.
4. Confirm the deletion by entering 1.

```
ADB2CONF  -- DB2X Delete accelerated table confirmation------------ 13:41

 Confirm the deletion of the accelerated table below.
 Schema: SALES
 Name  : SCADI901
 Accelerator : V1


 Select a choice
   1.  Continue with deletion
   2.  Cancel
```

Figure 347. Delete accelerated table confirmation panel (ADB2CONF)

Enabling and disabling automatic reload of accelerated tables

When you change data in an accelerated table, you can specify whether to automatically detect those changes and reload the accelerated table. This feature is useful when you insert, delete, or update records in an accelerated table.

Procedure

Select option P.CH on the Administration Menu panel.
The Options for Change Functions panel is displayed, as shown in the following figure:

```
ADB2PCO in                Options for Change Functions      11:26
Command ===>

Recreate accelerated tables . . . . . . .       (Yes/No. Default is Yes.)
 Reload accelerated tables . . . . . . . .        (Yes/No. Default is Yes.)
Add new DB2 tables to accelerator . . . .       (Yes/No. Default is Yes.)
Remove deleted tables from accelerator .        (Yes/No. Default is Yes)
Processing order  . . . . . . . . . . .         (T - object type, H - DB hierarchy.)
                                                  (Default is H.)
Statement validation exit name . . . . .         Name of EXEC used to validate
                                                   statments in WSL Validate
```

Figure 348. Options for Change Functions panel (ADB2PCO)

Enter Yes to mark DB2 accelerated tables to be eligible for automatic reload, or
enter No to specify that DB2 accelerated tables are not eligible for automatic reload.

Enabling and disabling incremental updates to accelerated tables

You can enable incremental updates to accelerated tables to automatically update
tables on an accelerator.

About this task

With incremental updates enabled, updates to tables are propagated to the
corresponding tables on the accelerator with little delay. Disabling incremental
updates excludes tables from the incremental update process.

Procedure

1. Select option AT on the System Administration panel.
2. Select option 1 on the Display/Manage Accelerated Tables panel.

   The Display Accelerated Tables panel is displayed, as shown in the following
   figure:

```
ADBPZAT n ----------- DB2X Display Accelerated Tables -------- Row 1 to 1 of 3
Command ===>                                              Scroll ===> PAGE

Commands: RTS  ADD  LOAD  ENABLE  DISABLE  DET
Line commands:
  I - Interpret  AC - Accelerator  T - Table  RTS - RTS info  L - Load
 AR - Archive  EN - Enable  DI - Disable  DE - Delete  DET - Table details
 ? - Show all line commands

    Table            Table     Server         Remote   Remote
S   Name             Schema    Name     E A   Name     Schema    Refresh Time
    *                *         *        * * *          *         *
--- ----------------> -------- -------- - - -------> -------- --------------->
    TBOC5I03         RAXESHP   V1       Y     TBOC5I03 RAXESHP  2013-08-21-06.28.
```

Figure 349. DB2 Display Accelerated Tables panel

3. On the Display Accelerated Tables panel, enter the EU line command to enable
   incremental updates, or enter the DU line command to disable incremental
   updates, as shown in the following figure:

```
ADBPZAT n ----------- DB2X Display Accelerated Tables -------- Row 1 to 5 of 5
Command ===>                                                Scroll ===> PAGE

Commands: RTS  ADD  LOAD  ENABLE  DISABLE  DET
Line commands:
  I - Interpret  AC - Accelerator  T - Table  RTS - RTS info  L - Load
 AR - Archive  EN - Enable  DI - Disable  DE - Delete  DET - Table details
 ? - Show all line commands

    Table               Table    Server         Remote    Remote
 S  Name                Schema   Name     E A   Name      Schema    Refresh Time
    *                   *        *        * * *           *         *
 --- ------------------> -------- -------- - - -------> -------- --------------->
 EU  EJBR1              SYSADM   REAL1     N   EJBR1-ID  SYSADM    2013-09-26-17.13
 DU  EJBR2              SYSADM   REAL1     N   EJBR2-ID  SYSADM    2013-09-26-16.01
     EJBR2              SYSADM   VIRTUAL1 Y   EJBR2_ID  SYSADM    2013-09-26-16.00
     T1                 SYSADM   REAL1     N   T1-ID_16  SYSADM    2013-09-27-07.17
     TEST12344574547459 SYSADM   REAL1     N   TEST1234  SYSADM    2013-09-27-07.12
 ****************************** END OF DB2 DATA ******************************
```

Figure 350. DB2 Display Accelerated Tables panel (ADBPZAT)

Archiving a table partition

You can reduce DB2 storage space by archiving table partitions to the IBM DB2
Analytics Accelerator.

About this task

When you archive a table partition to IBM DB2 Analytics Accelerator, DB2 stores
active data only and archive data is moved to the accelerator, thus reducing DB2
storage space. Archiving a table partition is valid only when you are using DB2
V10 or later.

You can select partitions to archive from a selected table on the following panels:
ADB21T, ADB21S, ADB21SP, and ADBPZAT. Error messages are displayed if a
table cannot be archived because of reasons such as:

- The table not an accelerated table,
- The table is not in a partition by range table space,
- The table has LOB or XML columns and cannot be archived,
- The table has a column that has a foreign key relationship.

The AR line command displays a panel that lists the partitions for the specified
table. You can select a single partition, all partitions, or enter your own partition
range. When the parts are selected and ready to archive, DB2 Admin calls the
stored procedure that archives the partitions to the accelerated table database.

This example uses the ADB21T panel.

To archive one or more partitions or a partition range:

Procedure

1. On the Tables, Views, and Aliases panel (ADB21T), select the table to be
   archived by specifying the AR line command and press Enter. The Archive
   accelerated table partition panel is displayed, as shown in the following figure.

```
ADBP1ARC  --------DB2X Archive accelerated table partition----- Row 1 to 3 of 3

Table schema . : DSN8500
Table name . . : DEPTTS


Commands: ALL  RESET  ARCHIVE

line commands:  S - Select part  D - Deselect part

Input partition range . . .                          (see help for details)


Sel Part  Archive Limit Key Value
--- ------ ------- -------------------------------------------------------------->
S      1      100
       2      200
       3      999
***************************** END OF DB2 DATA ******************************
```

Figure 351. Archive accelerated table partition panel (ADBP1ARC)

> **Note:** You can archive table partitions in batch to reduce wait time. You can run batch by specifying YES for Run Accelerator functions in batch on panel ADB2P2.

2. Issue the S line command on each partition you want to archive. Use the D line command to deselect a part. After you press Enter, the tables that are marked for archiving display Y in the Archive column.

   To specify a partition range, input the partitions to archive by using the same syntax as the SYSPROC.ACCEL_ARCHIVE_TABLES stored procedure. Use a colon (:) to specify a range. Use a negative number to specify partitions that start from the last partition. For example, -2 specifies the second-to-last partition.

   Here are some examples of valid partition range values:

   1, 2   Specifies partitions 1 and 2

   1, 2:3   Specifies partitions 1, 2, and 3

   1:2,3   Specifies partitions 1, 2, and 3

   -2   Selects the second to last partition

   -2:-1   Selects the second to last partition and the last partition

   -3,-1   Selects the third to last partition and the last partition

3. After all of the partitions have been specified, select ARCHIVE and press Enter: The following line commands are available:

   ALL   Select all of the partitions that are to be archived.

   RESET
   　　　　Deselect all of the partitions.

   ARCHIVE
   　　　　Process the archive request.

Viewing real-time status information for accelerated tables

You can view RUNSTATS and real-time status information for accelerated tables to help you decide whether to reload the accelerated table.

About this task

Information about the accelerated tables is stored in the pseudo-catalog table, SYSACCEL.SYSACCELERATEDTABLES. Each DB2 connection instance has one pseudo-catalog table.

Procedure

1. Select option AT on the System Administration panel.

   The Display/Manage Accelerated Tables panel is displayed, as shown in the following figure. You can specify filters for the tables you want to display by entering selection criteria.

```
ADBPZMAT ----------- DB2X Display/Manage Accelerated Tables ------------ 16:13
Option ===>  1


1 - Display accelerated tables                        DB2 System: DB2X
                                                       DB2 SQL ID: SYSADM


Enter display selection criteria.  Settings: LIKE operator;  Criteria not saved
Name . . . . . . .                            > Created by . .            >
Schema . . . . . .            >                 Accelerator . .           >
Enabled  . . . . .                              Archived  . . .
Created within . .
Altered within . .
Refreshed within .
```

Figure 352. DB2 Display/Manage Accelerated Tables panel (ADBPZMAT)

2. Select option 1 on the Display/Manage Accelerated Tables panel.

   The Display Accelerated Tables panel is displayed, as shown in the following figure:

```
ADBPZAT n ----------- DB2X Display Accelerated Tables -------- Row 1 to 2 of 2
Command ===>                                              Scroll ===> PAGE

Commands: RTS  ADD  LOAD  ENABLE  DISABLE  DET
Line commands:
  I - Interpret  AC - Accelerator  T - Table  R - RTS info  L - Load
 AR - Archive  EN - Enable  DI - Disable  DE - Delete  DET - Table details
 ? - Show all line commands

    Table               Table    Server        Remote   Remote
S   Name                Schema   Name     E A  Name     Schema   Refresh Time
    *                   *        *        * * *         *        *
--- ------------------ -------- -------- - - ------->  -------- --------------->
R   SALES               SCADI901 REAL1    N N  SALES_ID SCADI901 2013-08-01-10.38
    TBOC5I09            RAXESHP  V1       N    TBOC5I09 RAXESHP  2013-08-01-16.38
***************************** END OF DB2 DATA *****************************
```

Figure 353. Display Accelerated Tables panel (ADBPZAT)

3. From the Display Accelerated Tables panel, issue the R line command to display real-time statistics for a particular table. You can also issue the RTS primary command to display real-time statistics for all the tables on the panel. This example shows the results of issuing the R line command.

   The Real-Time Statistics for Table panel is displayed, as shown in the following figure:

```
ADB21SS n ---- DB2X Real-Time Statistics for Table  ---------- Row 1 to 1 of 1

Line commands:
 I - Info

     Table
Sel Space    TBname    Part    Ext    Nactive        Space Instance
    *        *         *       *      *              *        *
--- -------- -------- ------ ------ ----------- ------------- --------
I   DSN8S20D SALES        0      1          36           144        1
***************************** END OF DB2 DATA *****************************
```

Figure 354. Real-Time Statistics for Table panel (ADB21SS)

4. Issue the I line command to display more detailed RUNSTATS information for the table.

   The Interpretation of an Object in SYSTABLESPACESTATS panel is displayed, as shown in the following figure:

```
ADB21SSI  ---- DSNB Interpretation of an Object in SYSTABLESPACESTATS --- 16:10
Command ===>

Details for: DSN8D50A.DSN8S20D
                                                           More:     +
Table Name . . . . . . . : DEPTTS2        Table schema . . : DSN8500
Data base  . . . . . . . : DSN8D50A       Table space  . . : DSN8S20D
Partition  . . . . . . . : 0              Instance . . . . : 1
DBID . . . . . . . . . . : 513            PSID . . . . . . : 7
Rows or LOBs in TS . . . : 0
Bytes row data occupies  : 0
Active Pages . . . . . . : 36
Number of pages with rows: 0
Size (in KB) . . . . . . : 144            Extents  . . . . : 1
TS Statistics Updated at : 2013-04-18-14.39.38.418575
Drive type . . . . . . . : HDD            List Prefetch Ctl: <null> - Unknown
Last time that this row was updated  . . . : ?

Statistical data since last REORG or LOAD REPLACE
Timestamp of last LOAD REPLACE . . . . . . : ?
Timestamp of last REORG  . . . . . . . . . : 2013-04-18-14.02.24.096717
Records or LOBs inserted . . . . . . . . . : 0
Records or LOBs deleted  . . . . . . . . . : 0
Rows updated . . . . . . . . . . . . . . . : 0
Not perfectly chunked LOBs inserted  . . . : 0
Not well-clustered records inserted  . . . : 0
Number mass deletes or dropped tables  . . : 0
Overflow records created (near)  . . . . . : 0
Overflow records created (far) . . . . . . : 0
Net number of bytes added or removed . . . : 0
```

Figure 355. Real-Time Statistics for Table panel (ADB21SS)

The table name and table schema are displayed together with database and table space information. RUNSTATS information is based on table space.

Viewing accelerated table details

You can create reports that show details for each accelerated table, including change and archive information for the entire table or, if it's a partitioned table, for each part separately.

Before you begin

Viewing accelerated table details is valid only when you are using DB2 V10 or later.

Procedure

1. Select option Z - DB2 system administration on the DB2 Administration panel.
2. Select option AC on the System Administration panel.
3. On the DB2 Accelerators panel, enter the DET command.

   The DB2 Accelerated Table Details panel is displayed. The following panel is an example of a partitioned table.

```
ADBPD  ----------------- DB2X Accelerated Table Details -------------- 05:39
Command ===>

Commands: SAVE  ZOOM
_ Details for accelerated table (label): SCADI901.SALES

  Part info type . . . : BY_RANGE
  Column name  . . . . : COL1

_ Part no : 2
  Logical no  . . . . : 1
  Limit key value . . : 2011-10-31

  Change information  :
    Category  . . . . : RELOAD_RECOMMENDED
    Last load TS  . . : 2012-01-09T11:53:27.997141Z
    Type  . . . . . . : DataUpdated
    Shared tablespace : No
    Data size . . . . : 105 MB

  Archive information :
    Timestamp . . . . : 2012-01-09T11:53:27.997141Z
    Data size . . . . : 105 MB
    Backup image  . . : ARCHIVE.DA11.DB000022.CUSTOMER.P0003

_ Part no : 3
  Logical no  . . . . : 2
  Limit key value . . : 2011-11-31

  Change information  :
    Category  . . . . : RELOAD_REQUIRED
    Last load TS  . . : 2012-01-09T11:53:27.997141Z
    Type  . . . . . . : PartitionAddedOrRotated
    Shared tablespace : No
    Data size . . . . : 105 MB

_ Part no : 4
  Logical no  . . . . : 3
  Limit key value . . : 2011-12-31

  Change information  :
    Category  . . . . : UNKNOWN
    Last load TS  . . : 2012-01-09T11:53:27.997141Z
    Type  . . . . . . : DataUpdated
    Shared tablespace : Yes

_ Part no : 5
  Logical no  . . . . : 4
  Limit key value . . : 2012-01-31

  Change information  :
    Category  . . . . : NONE
    Last load TS  . . : 2012-01-09T11:53:27.997141Z
    Type  . . . . . . : NoChangeDetected
    Shared tablespace : Yes
.
.
.
```

Figure 356. DB2 Accelerated Table Details panel (ADBPD)

For non-partitioned tables, information is displayed for the entire table instead of the individual parts, as shown in the following figure:

```
ADBPD  ---------------- DB2X Accelerated Table Details -------------- 05:39
Command ===>

_ Details for accelerated table (label): SCADI901.SALES

   Change information  :
     Category  . . . . : RELOAD_RECOMMENDED
     Last load TS  . . : 2012-01-09T11:53:27.997141Z
     Type  . . . . . . : DataUpdated
     Shared tablespace : No
     Data size . . . . : 105 MB

   Archive information :
     Timestamp . . . . : 2012-01-09T11:53:27.997141Z
     Data size . . . . : 105 MB
     Backup image  . . : ARCHIVE.DA11.DB000022.CUSTOMER.P0003
```

Figure 357. DB2 Accelerated Table Details panel (ADBPD)

What to do next

You can also view accelerated table details by using the DET line command from the Display Accelerated Tables panel:

1. Select option AT on the System Administration panel.
2. Select either the DET primary command to display details for all of the accelerated tables, or specify the DET line command to display details for a particular accelerated table. Panel ADBPD is displayed as shown in Figure 356 on page 485.

Specifying or altering distribution and organizing keys

You can specify distribution or organizing keys for accelerated tables. Distribution keys distribute data across multiple nodes, and organizing keys sort table rows into blocks. Applying these keys to your tables can speed up query response times.

Procedure

1. Select option AT on the System Administration panel.

   The Display/Manage Accelerated Tables panel is displayed, as shown in the following figure. You can specify filters for the tables you want to display by entering selection criteria.

```
ADBPZMAT ----------- DB2X Display/Manage Accelerated Tables ------------ 16:13
Option ===>  1


1 - Display accelerated tables                      DB2 System: DB2X
                                                    DB2 SQL ID: SYSADM

Enter display selection criteria.  Settings: LIKE operator;  Criteria not saved
Name . . . . . . .                          > Created by . .           >
Schema . . . . . .            >               Accelerator . .          >
Enabled  . . . . .                            Archived  . . .
Created within . .
Altered within . .
Refreshed within .
```

Figure 358. DB2 Display/Manage Accelerated Tables panel (ADBPZMAT)

2. Select option 1 on the Display/Manage Accelerated Tables panel.

The Display Accelerated Tables panel is displayed, as shown in the following figure.

```
ADBPZAT n ----------- DB2A Display Accelerated Tables -------- Row 1 to 1 of 1
Command ===>                                              Scroll ===> PAGE

Commands: RTS  ADD  LOAD  ENABLE  DISABLE  DET
Line commands:
  I - Interpret  AC - Accelerator  T - Table  RTS - RTS info  L - Load
 AR - Archive  EN - Enable  DI - Disable  DE - Delete  DET - Table details
 ? - Show all line commands

    Table                Table    Server       Remote  Remote
S   Name                 Schema   Name    E A  Name    Schema   Refresh Time
    *                    *        *       * * *        *        *
--- -------------------- -------- -------- - - -------> -------- --------------->
    TB1                  SYSADM   AC2      Y   TB1-ID_4 SYSADM   2015-04-07-13.49
```

Figure 359. DB2 Accelerated Tables panel (ADBPZAT)

3. On the Display Accelerated Tables panel, issue the KEYS line command next to the table that you want to specify a distribution or organizing key for.

   The Accelerated Table - Keys panel is displayed, as shown in the following figure.

```
ADBPZAK n ---------- DB2X Accelerated Table - Keys ----------- Row 1 to 3 of 3
Command ===>                                              Scroll ===> CSR

Commands: SAVE  CANCEL

Status for table . . : SYSADM.TB1                                        >
Used disk space  . . : 0      (in MB)
Skew . . . . . . . . : 0.000   (0 to 1)   Organized percent . : 100.0

Line command:  Dn - Distribution sequence  On - Organizing sequence
               R - Remove column sequence

S  Column Name         D O Col Type   Length  Scale Nulls
   *                   * * *             *      * *
-- ------------------ - - -------- -------- ------ -----
   ACTNO                 SMALLINT      2        0 N
   ACTKWD                CHAR          6        0 N
   ACTDESC               VARCHAR      20        0 N
```

Figure 360. DB2 Accelerated Table - Key panel (ADBPZAK)

4. On the Accelerated Table - Keys panel, issue a line command to specify the relative position of the column in the distribution or organizing key.
   - Valid values for distribution keys are D1, D2, D3, and D4.
   - Valid values for organizing keys are O1, O2, O3, and O4.
5. Issue the SAVE command to save your changes.

Managing system parameters

You can use DB2 Admin to view, update, and load DB2 subsystem parameters.

DB2 Admin displays the currently active parameters and allows you to customize them for your environment. The changed parameters are stored as a new source for assembling the DSNZPARM module. You can assemble and link-edit the new source into a new DSNZPARM load module. The system parameter source and load modules are referred to here by the name DSNZPARM, although you can assign them your own names.

DB2 Admin provides the SET SYSPARM LOAD option to issue the command to load and activate the module. With DB2, you can load a new subsystem parameter module into storage while DB2 is active, which enables you to change certain operational parameters without stopping and starting DB2.

Restriction: Only dynamic parameters can be loaded using this feature.

To use DB2 Admin to manage system parameters, select option 2Z on the System Administration panel. The System Parameters panel is displayed, as shown in the following figure. Use the System Parameters panel to display the current parameters, to create a new parameter source file, to assemble and link-edit it, and to access the SET SYSPARM options.

Note: This functionality requires that DB2 SDSNLOAD data sets be allocated in linklist or STEPLIB. If you do not allocate DB2 SDSNLOAD data sets , you must use the DSNTIJUZ batch job process to assemble and linkedit the DSNZPARM module.

```
DB2 Admin ------------------ DB2X System Parameters -------------------- 07:57
Option ===>

   1 - Display Parameters/Generate DSNZPARM source     DB2 System: DB2X
   2 - Assemble and Linkedit DSNZPARM module           DB2 SQL ID: R148286
  3A - SET SYSPARM LOAD(          )
  3B - SET SYSPARM RELOAD
  3C - SET SYSPARM STARTUP

Output datasets:
  DSNZPARM Source  ===> JCL.CNTL(TEST)
  LinkEdit SYSLMOD ===> ADBV37.ISPLLIB(TEST)

  Assembly listing ===> ADB.ASM.LIST
  LinkEdit listing ===> ADB.LKED.LIST
  Optional Debug   ===> ADB.DEBUG.LIST

Input datasets:
  Assembly STEPLIB ===>
  Assembly SYSLIB  ===> JCL.CNTL
                   ===> 'SYS1.MACLIB'
                   ===>

  LinkEdit SYSLIB  ===> 'DB2X10.SDSNLOAD'
                   ===>
                   ===>
Options:
  Assembly         ===> ADATA,LIST(133),OBJECT
  Linkedit         ===> LIST,XREF,LET,RENT
```

Figure 361. System Parameters panel (ADB2Z2Z)

The following list provides an overview of the options and fields on that panel. See the online help for more extensive information.

1 — Display Parameters/Generate DSNZPARM source
Select this option if you want to view and optionally change the current parameters. If you want to change parameter values, you must specify an output data set and member. If no changes are made, the member is not written.

2 — Assemble and Linkedit DSNZPARM module
Select this option to assemble and link-edit the parameters module. Be sure to specify the output SYSLMOD data set name, because that is where the new load module is stored.

SET SYSPARM options
Use these options to easily execute the SET SYSPARM commands.

3A — SET SYSPARM LOAD
Select this option to load a new system parameter load module into storage.

3B — SET SYSPARM RELOAD
Select this option to reload the previous parameter load module into storage.

3C — SET SYSPARM STARTUP
Select this option to reload into storage the parameter load module used at subsystem startup.

Output data sets
Enter information pertaining to the output data sets that are used in creating the systems parameter data set and in the subsequent assemble and link-edit steps. Specify the output data set, DSNZPARM source because this is where the new source is written. When a new load module is created, you must specify the output SYSLMOD data set.

Input data sets
Enter information pertaining to additional input libraries and data sets used in the assembly and link-edit steps. You should specify the Assembly SYSLIB because this data set contains the DSNZPARM macros, such as DSN6SPRM, and DSN6LOGP. DB2 Admin accesses these macros to determine which parameters that exist for the subsystem.

Options
Specify options that you want in effect at assembly and link-edit time.

System Parameters — System Parameters panel

You can view and change the current system parameters.

To view and change the current parameters, select option 1 on the System Parameters panel. The System Parameters – System Parameters panel is displayed, as shown in the following figure.

```
DB2 Admin --------- DB2X System Parameters - System Parameters ---------- 07:59
Command ===>

                                              DB2 System: DB2X
                                              DB2 SQL ID: R148286
                                                     More:    +
    Storage Sizes and Connections
    Operator and DDF Functions
    Tracing and Data Installation
    Locking (IRLM)
    Active Log
    Archive Log
    Protection and Data Definition
    Stored Procedures
    Data Sharing Parameters
    Application Programming Defaults
    Other Parameters
    Restart Parameters
    Allow Explain during Autobind  . . . . . . . . . . . . . . YES (ABEXP   ) *
    Allow Autobind Operations . . . . . . . . . . . . . . .   YES (ABIND   ) *
    Archive Log Allocation Unit  . . . . . . . . . . . . . . . CYL (ALCUNIT ) *
    Copy 1 prefix  . . . . . .               DB2X.ARCHLOG1 (ARCPFX1 ) *
    Copy 2 prefix  . . . . . .               DB2X.ARCHLOG2 (ARCPFX2 ) *
    Archive Retention Period  . . . . . . . . . . . . . . .    31 (ARCRETN ) *
    Archive WTOR Routing codes . .                        1,3,4 (ARCWRTC ) *
    Issue WTOR before Archive Mounts . . . . . . . . . . . . YES (ARCWTOR ) *
    Read COPY2 Archives First  . . . . . . . . . . . . . . .  NO (ARC2FRST) *
    Plan Authorization cache size  . . . . . . . . . . . .  1024 (AUTHCACH) *
    Bind New Version . . . . . . . . . . . . . . . . . . . BINDADD (BINDNV  ) *
    Archive Dataset Blocksize  . . . . . . . . . . . . . . 28672 (BLKSIZE ) *
    IMS/BMP Timeout factor . . . . . . . . . . . . . . . . .    4 (BMPTOUT ) *
    Catalog Archive Datasets . . . . . . . . . . . . . . . . YES (CATALOG ) *
    ICF Catalog Name  . . . . . . . . . . . . . . . . . .   DB2X (CATALOG ) *
    Current Degree Special Register  . . . . . . . . . . . .    1 (CDSSRDEF) *
    System Checkpoint Frequency (LOGLOAD) . . . . . . . . 50000 (CHKFREQ ) *
    Compact Archive Logs . . . . . . . . . . . . . . . . . .  NO (COMPACT ) *
    Maximum Concurrent Remote Connections  . . . . . . .    128 (CONDBAT ) *
    Contract CT Long storage pool  . . . . . . . . . . . . .  NO (CONTSTOR) *
    Maximum Concurrent Allied Threads . . . . . . . . . .    300 (CTHREAD ) *
    DBA can create aliases,views . . . . . . . . . . . . . .  NO (DBACRVW ) *
    Database Protocol for 3-part names . . . . . . . . . .  DRDA (DBPROTCL) *
    Tape unit Deallocation Minutes . . . . . . . . . . . .     0 (DEALLCT ) *
    Tape Unit Deallocation Seconds  . . . . . . . . . . . .      (DEALLCT ) *
```

Figure 362. System Parameters — System Parameters panel (ADB2ZZMN)

The System Parameters — System Parameters panel displays a list of currently active DB2 system parameters. The top twelve lines, which have no parameter values to the right, are selection fields. When selected, a secondary panel is displayed that shows the parameters organized by category.

The selection fields are followed by the dynamic parameters in alphabetical order. Enter new values for any parameters by overwriting the existing value. Only those parameters identified by an asterisk (*) can be loaded dynamically using the SET SYSPARM command.

Restriction: This message can also be issued for parameters not on this panel, but whose value has changed as a result of the assembly. This situation might occur if DB2 maintenance was applied to the macro data sets, thereby changing the internal parameter values, and no interim subsystem recycle was performed.

System Parameters — Archive Log panel

The System Parameters — Archive Log panel is an example of a secondary panel that is displayed when one of the fields is selected from the System Parameters — System Parameters panel.

In this example, the category Archive Log was selected. The following figure shows the System Parameters — Archive Log panel.

```
  DB2 Admin ------------ DB2X System Parameters - Archive Log ------------- 08:18
  Command ===>

                                                    DB2 System: DB2X
                                                    DB2 SQL ID: R148286

    Dual Archive Logs  . . . . . . . . . . . . . . . . . . . . YES (TWOARCH )
    Timestamp Archive Log datasets . . . . . . . . . . . . . . EXT (TSTAMP  ) *
    Copy 1 prefix  . . . . . .                      DB2X.ARCHLOG1 (ARCPFX1 ) *
    Copy 2 prefix  . . . . . .                      DB2X.ARCHLOG2 (ARCPFX2 ) *
    Archive Log Allocation Unit  . . . . . . . . . . . . . . . CYL (ALCUNIT ) *
    Primary Space Allocation . . . . . . . . . . . . . .       200 (PRIQTY  ) *
    Secondary Space Allocation . . . . . . . . . . . . .       200 (SECQTY  ) *
    Catalog Archive Datasets . . . . . . . . . . . . . . . . . YES (CATALOG ) *
    Copy 1 Archive Log Device Type  . . . . . . . . . .      SYSDA (UNIT    ) *
    Copy 2 Archive Log Device Type  . . . . . . . . . .      SYSDA (UNIT2   ) *
    Archive Dataset Blocksize  . . . . . . . . . . . . . . . 28672 (BLKSIZE ) *
    Maximum Read Tape Units  . . . . . . . . . . . . . . . .     2 (MAXRTU  ) *
    Tape unit Deallocation Minutes . . . . . . . . . . . .       0 (DEALLCT ) *
    Tape Unit Deallocation Seconds  . . . . . . . . . . .         (DEALLCT ) *
    Maximum Archive Entries in BSDS . . . . . . . . . . . .   1000 (MAXARCH )
    Issue WTOR before Archive Mounts . . . . . . . . . . . . . YES (ARCWTOR ) *
    Archive Retention Period  . . . . . . . . . . . . . . .     31 (ARCRETN ) *
    Quiesce Period . . . . . . . . . . . . . . . . . . . . .     5 (QUIESCE ) *
    Compact Archive Logs . . . . . . . . . . . . . . . . . . . NO (COMPACT ) *
    Archive copy 1 Mass Storage Group Name  . . . . . . .         (MSVGP   )
    Archive copy 2 Mass Storage Group Name  . . . . . . .         (MSVGP2  )
    Limit Backout Processing During Restart . . . . . . . . AUTO (LBACKOUT)
    Restart Backout Limit  . . . . . . . . . . . . . . . . .     5 (BACKODUR)
    Read COPY2 Archives First  . . . . . . . . . . . . . . . . NO (ARC2FRST) *
    Offload  . . . . . . . . . . . . . . . . . . . . . . . . . NO (OFFLOAD )
    Single Volume DASD Archives  . . . . . . . . . . . . . . . NO (SVOLARC )
```

Figure 363. System Parameters — Archive Log panel (ADB2ZZAL)

Unrecognized Macro Parameters panel

DB2 Admin accesses SDSNMACS, the Assembly SYSLIB data set specified by the user, to determine which DSNZPARM parameters exist for this subsystem.

An unrecognized macro was encountered and is displayed in the Unrecognized Macro Parameters panel, as shown in the following figure.

```
  DB2 Admin -------- DB2X Unrecognized Macro Parameters Row 1 to 1 of 1
  Command ===>

  The following are parameters in the supplied macro in the SDSNMACS
  dataset but are not recognized by this function.  Values from the
  current subsystem parameters could not be obtained.  Any listed
  values are the default value for the macro. You may specify a new
  value for a parameter by over-typing the default. If the macro does
  not provide a default and a value is required, an assembly error may
  occur.

  Macro     Parameter  Default
  DSN6ARVP  SUPRHERO   JOE
  ************************* Bottom of data ***************************
```

Figure 364. Unrecognized Macro Parameters panel (ADB2ZZTL)

Displaying global variables and their authorizations

Global variables enable you to share relational data between SQL statements without the need for application logic to support the data transfer. You can display information about the global variables, which are defined in the SYSVARIABLES catalog table.

Procedure

1. Select option GV on the System Catalog panel.

   The Global Variables panel is displayed, as shown in the following figure.

```
ADBP1GV n --------------- DSNB Global Variables ---------------- Row 1 of 74
 Command ===> _____ Scroll ===> PAGE

 Line commands:
  I - Interpretation  A - Auth  GEN - Generate DDL  DDL - Object DDL
  CRE - Create  COM - Comment  ALT - Alter  DROP - Drop  DO - Dependent objects
  ? - Show all line commands
                                   Data     Max
 Select Schema   Name             Type     Length Scale Default Text
        *        *                *        *      * *
 ------ -------- ----------------- -------- <----- ------ -------------------->
 _____ RIP      INT              INTEGER    4      0
 _____ RIP      CH1              CHAR       1      0
 _____ ULVEMAN  INT              INTEGER    4      0
 _____ ULVEMAN  CH1              CHAR       1      0
 _____ ULVEMAN  TUJCHAR          CHAR      10      0 '1111111111'
 _____ ULVEMAN  TUJINT           INTEGER    4      0 121
 _____ ULVEMAN  TUJCH12DCD       CHAR       4      0 CURRENT DEGREE
 _____ ULVEMAN  TUJDEC52         DECIMAL    5      2
 _____ ULVEMAN  TUJFLOAT         FLOAT      8      0
 _____ ULVEMAN  TUJTS0           TIMESTMP   7      0
 _____ ULVEMAN  TUJTS2           TIMESTMP   8      2
 _____ ULVEMAN  TUJTS6           TIMESTMP  10      6
 _____ ULVEMAN  TUJTS12          TIMESTMP  13     12
 _____ ULVEMAN  TUJTZ12          TIMESTZ   15     12
 _____ ULVEMAN  TUJLVCH          VARCHAR 32704     0
 _____ ULVEMAN  TTJTS6           TIMESTMP  10      6 CURRENT TIMESTAMP
 _____ ULVEMAN  TUJDATE          DATE       4      0
 _____ ULVEMAN  TUJTIME          TIME       3      0
 _____ ULVEMAN  TUJCH12          CHAR      12      0 CURRENT DATE
 _____ ULVEMAN  TUJDATEDCD       DATE       4      0 CURRENT DEGREE
 _____ ULVEMAN  TUJVCH128DCS     VARCHAR  128      0 CURRENT SQLID
 _____ ULVEMAN  TUJVCH128DUSER   VARCHAR  128      0 USER
 _____ ULVEMAN  TUJVCH8DCAC      CHAR       8      0 CURRENT APPLICATION C
 _____ ULVEMAN  TUJVCH8DCMTTFO   CHAR       8      0 CURRENT MAINTAINED TA
 _____ S29168   SMI              SMALLINT   2      0
 _____ S29168   BI               BIGINT     8      0
 _____ S29168   INT              INTEGER    4      0
 _____ S29168   REAL             FLOAT      4      0
 _____ S29168   DOUBLE           FLOAT      8      0
 _____ S29168   DATE             DATE       4      0
 _____ S29168   TIME             TIME       3      0
 _____ S29168   CHAR_FBD         CHAR       8      0
 _____ S29168   VCH              VARCHAR    8      0
```

Figure 365. Global Variables panel (ADBP1GV)

The following fields are displayed on this panel:

Schema
    The schema of the global variable.

Name
    The name of the global variable.

Data Type
The name of the data type.

Max Length
The maximum length of the global variable.

Scale
The scale of the global variable.

Default Text
The text of the default value of the global variable.

If the text is truncated, type EXPAND on the primary command line, position the cursor on the default text field, and press Enter to display all of the text.

2. Issue the I line command on the row for the global variable that you want to display information about.

The Interpretation of an Object in SYSVARIABLES panel is displayed, as shown in the following figure.

```
ADBP1GVI  ------- DSNB Interpretation of an Object in SYSVARIABLES ------
Command ===>

Details for global variable : SYSIBMADM.GET_ARCHIVE
Schema. . . . : SYSIBMAD >  Name. . . . . : GET_ARCHIVE
Owner . . . . : SYSIBM      Ownertype . . : Auth ID

Type schema . . . . . . : SYSIBM
Type name . . . . . . . : CHAR
Maximum length. . . . . : 1
Scale . . . . . . . . . : 0
Default text  . . . . . : 'N'
Identifier  . . . . . . : 1
DB2 release created . . : P - DB2 V11
Date/time of creation . : 2012-07-20-15.05.30.475321
Source type . . . . . . : 0
CCSID . . . . . . . . . : 1208 - UNICODE Mixed
Default clause  . . . . : 1
Row ID for LOBs . . . . : A704E7065A79336E290401D0678001000000000000201
Internal environment  . : 0
IBM required  . . . . . : Y
Remarks . . . . . . . . :
```

Figure 366. Interpretation of an Object in SYSVARIABLES panel (ADBP1GVI)

The following fields are displayed on this panel:

Schema
The schema of the global variable.

Name
The name of the global variable.

Owner
The authorization ID of the owner of the global variable.

Ownertype
The type of owner:

L   The owner is a role.

blank
The owner is an authorization ID.

Type schema
> The schema name of the data type. For built-in data types this value is
> SYSIBM.

Type name
> The unqualified name of the data type.

Maximum length
> The maximum length of the global variable.

Scale
> The scale of the global variable.

Default text
> The text of the default value of the global variable.
>
> If the text is truncated, type EXPAND on the primary command line, position
> the cursor on the default text field, and press Enter to display all of the
> text.

Identifier
> The identifier of the global variable.

Identifier
> The identifier of the global variable.

DB2 release created
> The release of DB2 that was used to create the object.

Date/time of creation
> The date and time that the global variable was created.

Source type
> The source type:
>
> **0**   A built-in data type.
>
> *internal_ID*
> > A distinct type.

CCSID
> The CCSID of the global variable. The CCSID encoding scheme and
> character set.

Default clause
> The default clause that is specified for the global variable.

Row ID for LOBs
> The row ID values for the LOB columns in the SYSVARIABLES table.

Internal environment
> The internal environment identifier.

IBM required
> The origin of the row:
>
> **Y**   The row came from the basic machine-readable material (MRM) tape.
>
> **N**   The row did not come from the basic machine-readable material (MRM)
> tape.

Displaying global variable authorizations

You can display information about the users who grant privileges to global
variables, and information about the users who hold the privileges. You can also
display information about any plans that use the privileges.

About this task

Authorization information is stored in the SYSIBM.SYSVARIABLEAUTH catalog table.

Procedure

1. Select option AO on the System Catalog panel.

   Authorization options are displayed, as shown in the following figure.

```
ADB21 min -------------------- DSNB System Catalog -------------------- 16:17
 Option ===>


 Authorization options:                               DB2 System: DSNB
  OO - Object options                                 DB2 SQL ID: PEDRO
  GA - Storage group auths          PA - Plan authorizations
  DA - Database authorizations      LA - Collection authorizations
  SA - Table space authorizations   KA - Package authorizations
  TA - Table authorizations         HA - Schema authorizations
  VA - View authorizations          EA - User defined data type authorization
  CA - Column authorizations        FA - Function authorizations
  ZA - System authorizations        OA - Stored procedure authorizations
  UA - User authorizations          QA - Sequence authorizations
  RA - Resource authorizations      TR - Trusted contexts
  RO - Roles                        PM - Permissions
  CM - Column masks                 GVA - Global variable authorizations


 Enter standard selection criteria (Using a LIKE operator, criteria saved):
 Name . . . .             > Grantor . . .         >
 Owner . . . .       >     Grantee . . .       >
 In D/L/H . .              > Switch Catalog Copy . . . N   (N/S/C)
 And/or other selection criteria (option xC shows you columns for option x)
 Column . . .              > Operator . .       Value . .
```

Figure 367. System Catalog panel: authorization options (ADB21)

2. Select option GVA on the System Catalog panel.

   The Global Variable Authorizations panel is displayed, as shown in the following figure.

```
ADBPAGV n ----------- DSNB Global Variable Authorizations ---- Row 1 to 3 of 3
 Command ===>                                              Scroll ===> CSR

 Commands: REVOKE
 Line commands:                                                   W
  R - Revoke I - Interpret  GV - Global Variable               R R
                                                               E I
                      G                          H             A T
 Select Grantor  Grantee  T Schema   Name          G Timestamp  D E
       *        *        * *        *             * *          * *
 ------ -------- -------- - ------->  ---------------- - --------->  - -
       SYSADM   PUBLIC     SYSIBM   CLIENT_IPADDR    2012-12-21 Y
       SYSADM   PUBLIC     SYSIBMAD GET_ARCHIVE      2012-12-21 Y
       SYSADM   PUBLIC     SYSIBMAD MOVE_TO_ARCHIVE  2012-12-21 Y
 ****************************** END OF DB2 DATA ******************************
```

Figure 368. Global Variable Authorizations (ADBPAGV)

The following fields are displayed on this panel:

Grantor
      The authorization ID of the user who granted the privilege.

Grantee
The authorization ID of the user who holds the privilege or the name of the plan that uses the privilege.

GT Grantee type

blank
An authorization ID.

L A role.

P An application package. The grantee is a package if COLLID is not blank.

Schema
The schema name of the global variable.

Name
The unqualified name of the global variable.

HG The authorization level of the user who granted the privileges:

blank
Not applicable

E SECADM

G ACCESSCTRL

S SYSADM

T DATAACCESS

Timestamp
The time when the GRANT statement was run.

READ
The privilege to read the global variable:

blank
Not held.

G Read from GRANT.

Y Read without GRANT.

WRITE
The privilege to write the global variable:

blank
Not held.

G Write from GRANT.

Y Write without GRANT.

3. Issue the I line command on the row for the global variable that you want to display authorization information about.

The Interpretation of an Object in SYSVARIABLES panel is displayed, as shown in the following figure.

```
ADBP1GVI  ------- DSNB Interpretation of an Object in SYSVARIABLES ------ 12:31
 Command ===>


 Details for global variable : SYSIBMADM.GET_ARCHIVE
 Schema. . . . : SYSIBMAD >  Name. . . . . : GET_ARCHIVE
 Owner . . . . : SYSIBM      Ownertype . . : Auth ID

 Type schema . . . . . . : SYSIBM
 Type name . . . . . . . : CHAR
 Maximum length. . . . . : 1
 Scale . . . . . . . . . : 0
 Default text  . . . . . : 'N'
 Identifier  . . . . . . : 1
 DB2 release created . . : P - DB2 V11
 Date/time of creation . : 2012-07-20-15.05.30.475321
 Source type . . . . . . : 0
 CCSID . . . . . . . . . : 1208 - UNICODE Mixed
 Default clause  . . . . : 1
 Row ID for LOBs . . . . : A704E7065A79336E290401D0678001000000000000201
 Internal environment  . : 0
 IBM required  . . . . . : Y - Came from machine readable material
 Remarks . . . . . . . . :
```

Figure 369. Interpretation of an Object in SYSVARIABLES panel (ADBP1GVI)

The following fields are displayed on this panel:

Name
    The name of the global variable.

Owner
    The authorization ID of the owner of the global variable.

Ownertype
    The type of owner:

    L   The owner is a role.

    blank
        The owner is an authorization ID.

Type schema
    The schema name of the data type. For built-in data types this value is
    SYSIBM.

Type name
    The unqualified name of the data type.

Maximum length
    The maximum length of the global variable.

Scale
    The scale of the global variable.

Default Text
    The text of the default value of the global variable.

    If the text is truncated, type EXPAND on the primary command line, position
    the cursor on the default text field, and press Enter to display all of the
    text.

Identifier
    The identifier of the global variable.

DB2 release created
    The release of DB2 that was used to create the object.

Date/time of creation
The date and time that the global variable was created.

Source type
The source type:

0   A built-in data type.

internal_ID
     A distinct type.

CCSID
The CCSID of the global variable. The CCSID encoding scheme and character set.

Default clause
The default clause that is specified for the global variable.

Row ID for LOBs
The row ID values for the LOB columns in the SYSVARIABLES table.

Internal environment
The internal environment identifier.

IBM required
The origin of the row:

Y   The row came from the basic machine-readable material (MRM) tape.

N   The row did not come from the basic machine-readable material (MRM) tape.

Remarks
A character string about this global variable that was generated by using the COMMENT statement.

Granting global variable authorizations

You can grant privileges to users so that they can use global variables. You can also grant the authority to grant privileges to others. The Grant Variable Privileges panel guides you through the process without requiring you to know the syntax of the GRANT SQL statements.

About this task

To grant privileges to a global variable:

Procedure

1. Select option 5 on the Execute SQL Statements panel.

   The Grant or Revoke Privileges On Objects panel is displayed, as shown in the following figure.

```
ADB2G min ----------- DB2A Grant/Revoke Privileges On Objects --------- 13:21
Option ===>

                                              DB2 System: DB2A
GRANT                          REVOKE          DB2 SQL ID: SYSADM
  GG  - Storage group           RG  - Storage group
  GD  - Database                RD  - Database
  GS  - Table space             RS  - Table space
  GT  - Table or view           RT  - Table or view
  GC  - Column
  GP  - Plan                    RP  - Plan
  GL  - Collection              RL  - Collection
  GK  - Package                 RK  - Package
  GZ  - System privilege        RZ  - System privilege
  GR  - Buffer pool             RR  - Buffer pool
  GH  - Schema                  RH  - Schema
  GE  - Distinct type           RE  - Distinct type
  GF  - Function                RF  - Function
  GO  - Stored procedure        RO  - Stored procedure
  GJ  - JAR file                RJ  - JAR file
  GQ  - Sequence                RQ  - Sequence
  GGV - Global Variable         RGV - Global variable

  CP  - Copy privileges
```

Figure 370. Grant or Revoke Privileges On Objects panel (ADB2G)

2. Specify GGV in the **Option** field and press Enter. The Grant Variable Privileges panel is displayed, as shown in the following figure.

```
ADBPGGV n --------------- DSNB Grant Variable Privileges ---------------- 13:21
 Command ===> _____

  GRANT                                             DB2 SQL ID: SYSADM

  Select a privilege with a Y or G (to specify WITH GRANT OPTION).
    _ ALL
    _ READ
    _ WRITE

  ON VARIABLE
    Schema . . . _____  >
    Name . . . . _____  >

  TO . . . . . . _____
    _____  >
```

Figure 371. Grant Variable Privileges panel (ADBPGGV)

3. Enter Y in any of the ALL, READ, or WRITE fields. You can also enter G to specify the GRANT WITH option.
4. In the ON VARIABLE section, enter the schema and the name.
5. In the TO field, enter the user ID or a list of user IDs separated by commas, to which to grant access. You can also specify a role by providing the role keyword and a defined role name. For example: ROLE *groupadm*.
6. Press Enter to grant the selected privilege.

   For a detailed description of the GRANT and REVOKE statements, refer to the SQL Reference for your DB2 version.

Revoking global variable authorizations

You can revoke the authority that users have to grant privileges to global variables and you can revoke the privileges that users have to use global variables. The Revoke Variable Privileges panel guides you through the process without requiring you to know the syntax of the REVOKE SQL statements.

Before you begin

By reviewing the Revoke Impact report, you can view the effects of revoking an authorization before you actually revoke it.

Procedure

1. Select option AO on the System Catalog panel.
2. Select option GVA on the System Catalog panel.

   The Global Variable Authorizations panel is displayed, as shown in the following figure.

```
ADBPAGV n ----------- DSNB Global Variable Authorizations ---- Row 1 to 3 of 3
 Command ===>                                              Scroll ===> CSR

 Commands: REVOKE
 Line commands:                                                    W
  R - Revoke I - Interpret  GV - Global Variable                   R R
                                                                   E I
                        G                            H             A T
 Select Grantor  Grantee T Schema    Name            G Timestamp   D E
        *         *       * *         *               * *           * *
 ------ -------- -------- - -------> ---------------- - ---------> - -
        SYSADM   PUBLIC     SYSIBM   CLIENT_IPADDR      2012-12-21 Y
        SYSADM   PUBLIC     SYSIBMAD GET_ARCHIVE        2012-12-21 Y
        SYSADM   PUBLIC     SYSIBMAD MOVE_TO_ARCHIVE    2012-12-21 Y
 ***************************** END OF DB2 DATA ******************************
```

Figure 372. Global Variable Authorizations (ADBPAGV)

3. Issue the R line command against the global variable whose authorization you want to revoke. The Revoke Variable Privileges panel is displayed, as shown in the following figure.

```
ADBPRGV n -------------- DSNB Revoke Variable Privileges --------------- 05:57
 Command ===> _____

  REVOKE                                            DB2 SQL ID: ULVEMAN

  Enter any character in front of the privilege to revoke it from the user:

  _ ALL
  Y READ
  Y WRITE

  ON VARIABLE
    Schema . . . ULVEMAN >
    Name . . . . TEST             >
  FROM
    From . . . . X1                                                  >
  BY
    By . . . . . _____  >
  RESTRICT . . . ___   (Yes/No)

  Report Revoke Impacts . . . YES  (Yes/No)
```

Figure 373. Revoke Variable Privileges panel (ADBPRGV)

4. Specify the following options:
   a. Specify the privilege that you want revoked.
   b. Specify the information for the FROM, BY, and RESTRICT clauses and press Enter. For more information, see the *DB2 Command Reference* publication.

   A Change Management prompt is displayed that shows you the SQL REVOKE statement.

Reviewing the Revoke Impact report

Before you revoke a global variable, you can review the Revoke Impact report to determine how the authorizations and database objects will be affected by executing the revoke.

About this task

Restriction: You cannot revoke a privilege from a global variable if any of the following conditions exist:
- A function that is owned by the revokee references (READ or WRITE privilege) the specified global variable.
- A view that is owned by the revokee references (READ or WRITE privilege) the specified global variable.
- A trigger that is owned by the revokee references (READ or WRITE privilege) the specified global variable.
- A procedure that is owned by the revokee references (READ or WRITE privilege) the specified global variable.

Procedure

1. Select option AO on the System Catalog panel.
2. Select option GVA on the System Catalog panel. The Global Variable Authorizations panel is displayed, as shown in Figure 372 on page 500.
3. Issue the R line command against the global variable whose authorization you want to revoke and type Yes in the **Report Revoke Impacts** field.

   The Revoke impact report is displayed as shown in the following figure.

```
ADB2RIP n -------------- DSNB Revoke Impact Report ---------------- Row 1 of 1
 Command ===> _____ Scroll ===> PAGE



 Line commands:  I - Interpretation
                                Owner/
 S    Grantee  G Resource N/ O  Schema/  Grantor/ G H Privileges/
   Lv          T Collection  T  P/K Name Binder   T G Effect
 - -- -------- - ----------- -- -------- -------- - - -------------------------
 _ 0  X1         TEST           GV ULVEMAN  ULVEMAN      YY
 ****************************** END OF DB2 DATA ********************************
```

Figure 374. Revoke Impact report (ADB2RIP)

The following fields are displayed on this panel:

Lv The cascade level, which represents the number of implied revokes that would lead to the revoke at the current line.

A value of 99 indicates that the level is 99 or higher.

Grantee
Authorization ID of the user who holds the privilege.

GT Grantee type:

Blank
Authorization ID

L Role

Resource N/ Collection
For most database objects or resources the column contains the object's name. For packages it contains the package's collection ID.

OT The character code that represents the database object type:

G Storage Group

D Database

S Table Space

T Table

P Plan

K Package

L Collection

E Distinct Type

B Buffer Pool

Z System

H Schema

F User-Defined Function

O Stored Procedure

GV Global Variable

Owner/ Schema/ P/K Name
For most objects, the column contains the object's owner ID, schema name, or database name. For plans and packages, it contains the name of the plan or package.

Grantor/ Binder
> For most objects, the column contains the authorization ID of the user who granted the privilege. For invalidated or inoperative plans or packages, it contains the user who did the bind.

GT Grantor type:

> **Blank**
> > Authorization ID
>
> **L** Role

HG Authorization level of the user from whom the privileges were received:

> **C** DBCTL
>
> **D** DBADM
>
> **M** DBMAINT
>
> **S** SYSADM
>
> **L** SYSCTRL
>
> **Privileges/ Effect**
> > The description of the privilege, a series of authorization characters, or the effect on the database object.

4. Issue the I line command on the row for the global variable that you want to display interpretation information for.

   The Interpretation of revoked privileges panel is displayed, as shown in the following figure.

```
ADB2RIPI  ---------- DSNB Interpretation of revoked privileges ---------- 07:34
 Command ===> _____

 Variable privileges:

 Variable schema . . . : ULVEMAN
 Variable name . . . . : CH1
 Held by auth ID . . . : RIPA
 Granted by  . . . . . : ULVEMAN
 Grant timestamp . . . : 2013-04-08-04.28.07.407623
 Auth level of grantor :

 The following privileges are held by the grantee:
 READ variable  . . :     Grant:
 WRITE variable . . : Yes Grant: No
```

Figure 375. Interpretation of revoked privileges panel (ADB2RIPI)

Displaying buffer pool status

You can display the current status of one or more active or inactive buffer pools.

About this task

To display the current status of one or more active or inactive buffer pools:

Procedure

1. Select option BD on the System Administration panel. The Display Buffer Pools panel is displayed, as shown in the following figure.

```
DB2 Admin ----------------- DB2X Display Buffer Pools ----------------- 16:07
Command ===>

 -DISPLAY BUFFERPOOL(
  Buffer pool name    ===>           (Active, BP0-49, BP8K_, BP16K_, BP32K_, *)
) DETAIL(
  Include details     ===>           (Interval or *)
) LIST(
  Include page sets   ===>           (Active or *)
) LSTATS
  Page set statistics ===>           (Yes/No)

  Max DB2 output (KB) ===> 32        (1-1000)
```

Figure 376. Display Buffer Pools panel (ADB2ZBD)

2. Enter the appropriate keywords and parameters on the panel. DB2 Admin issues the DB2 -DISPLAY BUFFERPOOL command. The information that DB2 Admin returns to you from the command is in ISPF browse format.

Altering buffer pools

You can alter the attributes of active or inactive buffer pools.

About this task

To alter the attributes of active or inactive buffer pools:

Procedure

1. Select option BA on the System Administration panel. The Alter Buffer Pools panel is displayed, as shown in the following figure.

```
ADB2ZBA2  --------------- Alter Buffer Pools ---------- Row 1 to 49 of 80
Command ===>                                            Scroll ===> CSR

Line commands:
 AL - Alter buffer pool  DIS - Display buffer pool

     BP    VP  VPSZ VPSZ FM PG     VP   VP PG         Int1  Int2  VP X Auto
Sel Name  Size Min  Max  SZ Steal SEQT PSEQT FIX DWQT VDWQT VDWQT PSEQT Size
     *     *    *    *   * * *      *    * *   *    *    *     *     * *
--- ------ ---> ---> ---> -- ----- ----- ----- --- ----- ----- ----- ----- ----
    BP0   2000 0    0    1M LRU    80   50 NO   30   5     0     0 YES
    BP1   2000 1000 3000 1M LRU    80   50 NO   30   5     0     0 YES
    BP2   2000 2002 2000 1M LRU    80   50 NO   30   5     0     0 NO
    BP3   0    0    0    4K LRU    80   50 NO   30   5     0     0 NO
    BP4   1000 0    0    4K LRU    80   50 NO   30   5     0     0 NO
    BP5   0    0    0    4K LRU    80   50 NO   30   5     0     0 NO
    BP6   0    0    0    4K LRU    80   50 NO   30   5     0     0 NO
    BP7   0    0    0    4K LRU    80   50 NO   30   5     0     0 NO
    BP8   0    0    0    4K LRU    80   50 NO   30   5     0     0 NO
    BP9   0    0    0    4K LRU    80   50 NO   30   5     0     0 NO
    BP10  0    0    0    4K LRU    80   50 NO   30   5     0     0 NO
    BP11  1000 0    0    4K LRU    80   50 NO   30   5     0     0 NO
    BP12  0    0    0    4K LRU    80   50 NO   30   5     0     0 NO
    BP13  0    0    0    4K LRU    80   50 NO   30   5     0     0 NO
```

Figure 377. Alter Buffer Pools panel (ADB2ZBA2)

The following fields are available on this panel:

SELECT
    Input field where you enter one of the line commands listed on the panel.

BP NAME
　　Buffer pool name.

VP SIZE
　　Virtual buffer pool size.

VPSZ MIN
　　The minimum size for the buffer pool.

VPSZ MAX
　　The maximum size for the buffer pool.

FM SZ
　　The frame size for the buffer pool.

HP SIZE
　　Hiperpool size.

CAST OUT
　　Hiperspace* CASTOUT value.

VP SEQT
　　Virtual sequential steal threshold.

VP PSEQT
　　Virtual parallel sequential threshold.

HP SEQT
　　Hiperpool sequential steal threshold.

DWQT
　　Deferred write threshold.

VDWQT
　　Vertical deferred write threshold.

VP X PSEQT
　　Assisting virtual parallel sequential threshold.

Auto Size
　　Specifies whether the buffer pool adjustment is turned on or off.

　　NO　　Specifies that the buffer pool does not use Workload Manager
　　　　　　(WLM) services for automatic buffer pool sizing adjustment. This is
　　　　　　the default.

　　YES　　Specifies that the buffer pool uses WLM services, if available, to
　　　　　　automatically adjust the size of the buffer pool based on dynamic
　　　　　　monitoring of the workload goals and the available storage on the
　　　　　　system.

2. Issue one of the following line commands:
   - AL to alter a buffer pool. When you press Enter, DB2 Admin issues the
     -ALTER BUFFERPOOL command.
   - DIS to display buffer pool. When you press Enter, DB2 Admin issues the
     -DISPLAY BUFFERPOOL command.

   The information DB2 Admin returns to you from the commands is in ISPF
   browse format.

Displaying buffer pool hit ratios

You can name the buffer pools for which buffer pool hit ratios should be
displayed.

About this task

The hit ratio is calculated as the number of hits in the buffer pool divided by the number of GETPAGES.

Procedure

1. Select option BH on the System Administration panel. The Display Buffer Pool Hit Ratios panel is displayed, as shown in the following figure.

```
DB2 Admin -------------- DB2X Display Buffer Pool Hit Ratios ------------ 23:45
Command ===>

 -DISPLAY BUFFERPOOL(
  Buffer pool name    ===>           (Active, BP0-49, BP8K_, BP16K_, BP32K_, *)
) DETAIL(
  Include details     ===>           (Interval or *)
)
```

Figure 378. Display Buffer Pool Hit Ratios panel (ADB2ZBH)

2. Enter the name of a buffer pool. The following values are valid:

   Active All active buffer pools.

   BP0–BP49, BP8K_, BP16K_, BP32K_
   Select a specific buffer pool name from the valid values available.

   *   All buffer pools.
3. Specify the interval for which information should be displayed; the interval can be either since the buffer pool was created (*) or since the last display (interval).
4. Press Enter. DB2 Admin issues the DB2 DISPLAY BUFFERPOOL command to generate the Buffer Pool Hit Rations panel, as shown in the following figure.

```
DB2 Admin -------------- DB2X Buffer Pool Hit Ratios --------------------------
Command ===>

Line commands:  DIS  - Display buffer pool

        BP                         Random    Random    Hit
Select Name   VP Size  HP Size Get Pages      I/Os    Ratio
------ ------ -------- ------- --------- ---------- ------
       BP0                        63605       1262  98.02
       BP1                          256         14  94.53
       BP2                          568         99  82.57
       BP3                          519         12  97.69
       BP32K                       1152          0 100.00
       BP8K0                      38772       2134  94.50
       BP16K0                       556         12  97.84
***************************** END OF DB2 DATA ******************************
```

Figure 379. Buffer Pool Hit Ratios panel (ADB2ZBH2)

The following fields are available on this panel:

SELECT
   Input field where you list one of the line commands listed on the panel.

BP NAME
   Name of the buffer pool.

VP SIZE
   Size of the virtual buffer pool.

HP SIZE
Size of the hiperpool.

RANDOM GET PAGES
Number of random GETPAGES (RGP).

RANDOM I/Os
Number of random I/Os (RIO).

HIT RATIO
Buffer pool hit ratio, which is calculated as follows:

```
100 * (RGP - RIO) / RGP
```

Viewing group buffer pools

You can view buffer pools that are in DB2 data sharing.

Procedure

1. Select option **GD** on the System Administration panel. The Display Group Buffer Pool panel (ADB2ZGD) is displayed, as shown in the following figure.

```
ADB2ZGD--------------- DB2X Display Group Buffer Pools --------------- 23:43
Command ===>

 -DISPLAY GROUPBUFFERPOOL
  Name . . . . . . . . . gbp0     >  (GBP0-49, GBP8K0-9, GBP16K0-9, GBP32K-9
                                      or structure name)
  TYPE . . . . . . . . .            (G - GCONN, M - MCONN, N - NOCACHE, or *)
  MDETAIL  . . . . . . .            (I - INTERVAL, or *)
  GDETAIL  . . . . . . .            (I - INTERVAL, or *)
  CONNLIST . . . . . . .            (Yes/No)
  Max DB2 output (KB)  . 32         (1-9999)
```

Figure 380. Display Group Buffer Pool panel (ADB2ZGD)

The following fields are available on this panel:

NAME
Group buffer pool name.

TYPE
Specifies the type of group buffer pools.

> **GCONN**
> Group buffer pools that are currently connected to any member of the data sharing group.

> **MCONN**
> Group buffer pools that are currently connected to the member to which the command is directed.

> **NOCACHE**
> Group buffer pools that have the GBPCACHE attribute set to NO.

MDETAIL
Shows a detailed statistical report that lists the member's activity for each group buffer pool. If a group member has never been actively connected to the group buffer pool, no detail report is shown. The default is interval, which means the report shows incremental statistics.

GDETAIL
Shows a detailed statistical report that lists the activity of the entire group

for each group buffer pool. If a group member is not actively connected to the group buffer pool, no detail report is shown.

CONNLIST
Specifies whether a connection list report is shown for the specified group buffer pools. The report lists the connection names of the subsystems that are currently connected to the group buffer pools and provides connection status.

Max DB2 output
Specifies the maximum size of ISPF table that stores the report for the group buffer pool.

2. Optional: Press Enter to run the **DISPLAY GROUPBUFFERPOOL** command. The Browse DB2 Command Output panel (ADB2DB2O) is displayed, as shown in the following figure.

```
ADB2DB2O------ DB2X Browse DB2 Command Output ---  Line 00000000 Col 001 080
 Command ===>                                              Scroll ===> CSR

 -DISPLAY GROUPBUFFERPOOL(GBP0)


******************************* Top of Data **********************************
DSNB750I  @ DISPLAY FOR GROUP BUFFER POOL GBP0 FOLLOWS
DSNB755I  @ DB2 GROUP BUFFER POOL STATUS
           CONNECTED                               = YES
           CURRENT DIRECTORY TO DATA RATIO         = 5
           PENDING DIRECTORY TO DATA RATIO         = 5
           CURRENT GBPCACHE ATTRIBUTE              = YES
           PENDING GBPCACHE ATTRIBUTE              = YES
DSNB756I  @  CLASS CASTOUT THRESHOLD                = 5%
           GROUP BUFFER POOL CASTOUT THRESHOLD     = 30%
           GROUP BUFFER POOL CHECKPOINT INTERVAL   = 4 MINUTES
           RECOVERY STATUS                         = NORMAL
           AUTOMATIC RECOVERY                      = Y
DSNB757I  @ MVS CFRM POLICY STATUS FOR DSNCAT_GBP0   = NORMAL
           MAX SIZE INDICATED IN POLICY            = 8196 KB
           DUPLEX INDICATOR IN POLICY              = DISABLED
           CURRENT DUPLEXING MODE                  = SIMPLEX
           ALLOCATED                               = YES
DSNB758I  @   ALLOCATED SIZE                        = 6144 KB
           VOLATILITY STATUS                       = VOLATILE
           REBUILD STATUS                          = NONE
           CFNAME                                  = LF01
           CFLEVEL - OPERATIONAL                   = 14
           CFLEVEL - ACTUAL                        = 14
DSNB759I  @   NUMBER OF DIRECTORY ENTRIES           = 4667
           NUMBER OF DATA PAGES                    = 930
           NUMBER OF CONNECTIONS                   = 2
DSNB798I  @ LAST GROUP BUFFER POOL CHECKPOINT
                                  00:27:48 AUG 12, 2013
           GBP CHECKPOINT RECOVERY LRSN            = CBCCB4A0D113
           STRUCTURE OWNER                         = VA1B
DSNB790I  @ DISPLAY FOR GROUP BUFFER POOL GBP0 IS COMPLETE

****************************** Bottom of Data ********************************
```

Figure 381. Browse DB2 Command Output panel (ADB2DB2O)

Altering group buffer pools

You can alter the information for group buffer pools that are in DB2 data sharing.

Procedure

1. Select option **GA** on the System Administration panel. The Alter Group Buffer Pools panel (ADBPZGA2) is displayed, as shown in the following figure.

```
ADBPZGA2  ------------ DB2X Alter Group Buffer Pools ------- Row
Command ===>                                              Scr

Line commands:
 AL - Alter buffer pool  DIS - Display buffer pool

     GBP      GBP                                  GBP  GBP
Sel Name     Cache Autorec Ratio Classt1 Classt2  Poolt Chkpt
     *        *     *       *     *       *        *    *
--- -------- ----- ------- ----- ------- -------- ----- ------
    GBP0     YES   Y          5      89    32766    30      4
    GBP1     YES   Y          5       5     1000    30      4
    GBP2     YES   Y          5       5    10000    30      4
    GBP3     YES   Y          5       5        0    30      4
    GBP4     YES   Y          5       5        0    30      4
    GBP5     YES   Y          5       5        0    30      4
    GBP6     YES   Y          5       5        0    30      4
    GBP7     YES   Y          5       5        0    30      4
    GBP8     YES   Y          5       5        0    30      4
    GBP9     YES   Y          5       5        0    30      4
    GBP10    YES   Y          5       5        0    30      4
```

Figure 382. Alter Group Buffer Pools panel (ADBPZGA2)

The following fields are available on this panel:

SELECT
Input field where you enter one of the line commands that are listed on the panel.

GBPName
Group buffer pool name.

GBPCache
Shows the pending group buffer pool cache attribute. The value **Yes** indicates that the group buffer pool is used for both caching and cross-invalidation.

Autorec
Indicates whether automatic recovery is specified for the group buffer pool.

Classt1
Shows a percentage that indicates the degree to which data entries fill the data pages in the group buffer pool.

GBPPoolt
Displays the castout threshold for a group buffer pool. When the threshold is met, the data in the group buffer pool is cast out to disk.

GBPChkpt
Shows the checkpoint interval for a group buffer pool.

2. Choose one group buffer pool and type the line command **AL** in the **Sel** column. The Alter Group Buffer Pools panel (ADBPZGA8) is displayed, as shown in the following figure.

```
ADBPZGA8  ---------------- DB2A Alter Group Buffer Pool ---------------- 23:57
Command ===>

 -ALTER GROUPBUFFERPOOL
  Name . . . . . . . . . . GBP3     >  (GBP0-49, GBP8K0-9, GBP16K0-9, GBP32K-9
                                        or structure name)
  GBPCACHE . . . . . . . . YES        (Yes/No)
  AUTOREC  . . . . . . . . Y          (Yes/No)
  RATIO  . . . . . . . . . 5          (1.0-255)
  CLASST1  . . . . . . . . 5          (0-90)
  CLASST2  . . . . . . . . 0          (0-32767)
  GBPOOLT  . . . . . . . . 30         (0-90)
  GBPCHKPT . . . . . . . . 4          (1-999999)
```

Figure 383. Alter Group Buffer Pools panel (ADBPZGA8)

3. Optional: Change the group buffer pool parameters.

The following fields are available on this panel for you to alter:

Name
> Group buffer pool name.

GBPCache
> Specifies whether group buffer pool is to be used for both caching data and cross-invalidation, or just for cross-invalidation.

Autorec
> Specifies whether automatic recovery by DB2 takes place when a structure failure occurs, or when the connectivity to all members of the group buffer pool is lost

Classt1
> A percentage of the number of data entries and can be an integer 0 - 90, inclusive. The default is 5.

Classt2
> An absolute number of pages.

GBPPoolt
> The threshold at which data in the group buffer pool is cast out to disk.

GBPChkpt
> Changes the time interval, in minutes, between successive checkpoints of the group buffer pool.

4. Press Enter to run the **ALTER GROUPBUFFERPOOL** command. The Statement Execution Prompt panel is displayed.

```
ADB2PSTM  --------------- DB2A Statement Execution Prompt --------------- 23:59
Option ===> 1

DB2 Admin is about to execute the statement below. You have asked to be
prompted before DB2 Admin executes this type of statement. What do you want to
do now:
   1 - Execute the statement
   2 - Edit the statement
   3 - Create a batch job with the statement
   4 - Add the statement to the work statement list
 CAN - Cancel
 Work statement list dsn  ===> 'SYSADM.AANECM.WSL'
 Work statement list name ===> C0000001  Action ===> A (Append or Replace)
                                                            More:     +
Statement that is about to be executed (first 28 lines):
-ALTER GROUPBUFFERPOOL(GBP3) GBPCACHE(NO) AUTOREC(NO) RATIO(3.14) CLASST
(55,22222) GBPOOLT(66) GBPCHKPT(149527)
```

Figure 384. Statement Execution Prompt panel (ADB2PSTM)

5. After the command runs, return to the Alter Group Buffer Pools panel
   (ADBPZGA2) to see the changes that you made.

Displaying archive log information

You can display information about the input archive log.

About this task

To display information about the input archive log:

Procedure

Select option LD on the System Administration panel. The Display Archive Log
Parameters panel is displayed, as shown in the following figure.

```
 DB2 Admin ------- DB2X Browse DB2 Command Output ---  Line 00000000 Col 001 080
 Command ===>                                              Scroll ===> PAGE

  -DIS ARCHIVE


******************************* TOP OF DATA  ********************************
DSNJ322I Û DISPLAY ARCHIVE REPORT FOLLOWS-
               COUNT                 TIME
             (TAPE UNITS)         (MIN,SEC)
DSNZPARM         2                    0,00
CURRENT          2                    0,00
===============================
ADDR STATUS CORR-ID  VOLSER DATASET_NAME
NO TAPE ARCHIVE READING ACTIVITY.
END OF DISPLAY ARCHIVE REPORT.
DSN9022I Û DSNJC001 '-DIS ARCHIVE' NORMAL COMPLETION
****************************** BOTTOM OF DATA *******************************
```

Figure 385. Display Archive Log panel (ADB2DB2O)

DB2 Admin generates this panel by issuing the -DISPLAY ARCHIVE command.

Setting archive log parameters

You can set the upper limit for the number of and the deallocation time of tape
units for the archive log.

About this task

To set the upper limit for the number of and the deallocation time of tape units for the archive log:

Procedure

1. Select option LS on the System Administration panel. The Set Archive Log Parameters panel is displayed, as shown in the following figure.

```
DB2 Admin --------------- DB2X Set Archive Log Parameters --------------- 16:08
Command ===>

 -SET ARCHIVE

  COUNT(
  Max tape units      ===> 2        (1-99, DSNZPARM default is     2)

) TIME(
  Tape retain minutes ===> 0        (0-1440, DSNZPARM default is   0)
  Tape retain seconds ===> 00       (0-59)
)
```

Figure 386. Set Archive Log Parameters panel (ADB2ZLSS)

2. Enter the appropriate keywords and parameters on the panel. Enter the following values:
   - Max tape units
   - Tape retain minutes
   - Tape retain seconds
3. Press Enter. DB2 Admin issues the DB2 -SET ARCHIVE command with the parameter settings that you specified. The information DB2 Admin returns to you from the command is in ISPF browse format.

Archiving the current DB2 log

You can archive the current DB2 log.

About this task

To archive the current DB2 log:

Procedure

1. Select option LA on the System Administration panel. The Archive Current Log panel is displayed, as shown in the following figure.

```
DB2 Admin ----------------- DB2X Archive Current Log ------------------- 16:08
Command ===>

 -ARCHIVE LOG

  MODE(QUIESCE)
  Create system POC    ===>                (Yes/No)
  TIME(
  Max POC quiesce secs ===>                (1-999)
) WAIT(
  Wait for POC         ===>                (Yes/No)
)
```

Figure 387. Archive Current Log panel (ADB2ZLA)

2. Enter the appropriate keywords and parameters on the panel and press Enter. DB2 Admin issues the DB2 -ARCHIVE LOG command. The command response that DB2 Admin returns to is displayed in an ISPF browse session.

Displaying log information

You can display information about the DB2 log.

About this task

To display information about the DB2 log:

Procedure

1. Select option LI on the System Administration panel. The Display Log Information panel is displayed, as shown in the following figure.

```
DB2 Admin ------- DB2X Browse DB2 Command Output ---  Line 00000000 Col 001 080
Command ===>                                                  Scroll ===> PAGE

 -DIS LOG


******************************* Top of Data **********************************
DSNJ370I DB2X DSNJC00A LOG DISPLAY
CURRENT COPY1 LOG = DB2X.LOGCOPY1.DS02 IS 75% FULL
CURRENT COPY2 LOG = DB2X.LOGCOPY2.DS02 IS 75% FULL
H/W RBA = 000003AF8836, LOGLOAD = 50000
FULL LOGS TO OFFLOAD = 0 OF 6, OFFLOAD TASK IS (AVAILABLE)
DSNJ371I DB2X DB2 RESTARTED 19:45:59 NOV 28, 2003
RESTART RBA 000003AC7000
DSN9022I DB2X DSNJC001 '-DIS LOG' NORMAL COMPLETION
****************************** Bottom of Data ********************************
```

Figure 388. Display Log Information panel (ADB2DB2O)

2. Enter the appropriate keywords and parameters and press Enter. DB2 Admin issues the DB2 -DISPLAY LOG command. The information DB2 Admin returns to you from the command is in ISPF browse format.

Changing DB2 system checkpoint frequency

You can change how frequently DB2 should perform a system checkpoint.

About this task

To change how frequently DB2 should perform a system checkpoint (in terms of number of DB2 log records):

Procedure

1. Select option LZ on the System Administration panel. The Change DB2 System Checkpoint Frequency panel is displayed, as shown in the following figure.

```
DB2 Admin -------- DB2X Change DB2 System Checkpoint Frequency ---------- 16:51
Command ===>

 -SET LOG
 Mode  . . . . . . . . . .            (S-SINGLE, B-BOTH)
 LOGLOAD(
 Checkpoint frequency  . .           (1000-16000000 when Mode=S,
 )                                     0 or 1000-99999999 when Mode=B)

 CHKTIME(
 Checkpoint frequency  . .           (1-60   when Mode=S,
 )                                     0-1439 when Mode=B)

 SUSPEND . . . . . . . . .           (Yes/No)
 RESUME  . . . . . . . . .           (Yes/No)
 NEWLOG  . . . . . . . . .
 COPY  . . . . . . . . . .           (1/2)
```

Figure 389. Change DB2 System Checkpoint Frequency panel (ADB2ZLZ)

2. Enter the appropriate keywords and parameters on the panel and press Enter.
   DB2 Admin issues the DB2 -SET LOG command. The information DB2 Admin
   returns to you from the command is in ISPF browse format.

Displaying or updating communications settings

DB2 uses communication settings that you can display or update.

About this task

These settings are stored in communication database (CDB) tables (SYSIBM.xxx).

Procedure

1. Select option DU on the System Administration panel. The Display/Update
   CDB panel is displayed, as shown in the following figure.

```
DB2 Admin ------------------ DB2X Display/Update CDB ------------------- 17:34
Option ===>

                                                 DB2 System: DB2X
   L - Display/update LOCATIONS                  DB2 SQL ID: ISXSTL
   1 - Display/update LUNAMES
   2 - Display/update IPNAMES
   3 - Display/update LUMODES
   4 - Display/update MODESELECT
   5 - Display/update USERNAMES
   6 - Display/update LULIST
   7 - DISPLAY/UPDATE IPLIST

Option xI can be used to insert rows into empty tables (x= option number)

Switch Catalog Copy  ===>  N   (N/S/C)
```

Figure 390. Display/Update CDB panel (ADB2Z5)

2. Select one of the following options and press Enter. Another panel is displayed
   that lists the rows in the corresponding CDB table.

   - Select option L to delete, insert, or update rows in the SYSIBM LOCATIONS
     table.
   - Select option 1 to delete, insert, or update rows in the SYSIBM LUNAMES
     table.

- Select option 2 to delete, insert, or update rows in the SYSIBM IPNAMES table.
- Select option 3 to delete, insert, or update rows in the SYSIBM LUMODES table.
- Select option 4 to delete, insert, or update rows in the SYSIBM MODESELECT table.
- Select option 5 to delete, insert, or update rows in the SYSIBM USERNAMES table.
- Select option 6 to delete, insert, or update rows in the SYSIBM LULIST table.
- Select option xI (where x represents one of the previous seven option identifiers) to insert rows into an empty CDB table. For example, to insert rows into the SYSIBM.LVMODES table, enter 3I.

3. Follow the directions on the panel that is displayed.

Results

Use this panel to select the table in the communications database (CDB) you want to display or update.

If you want to insert rows into an empty table, you can do this by choosing option xI, where x represents the table (for example, 3I tells DB2 Admin to insert rows into the LUMODES table).

Displaying or updating the LOCATIONS table

Use the Display/Update LOCATIONS table to update the LOCATIONS table.

Select option L on the Display/Update communications database (CDB) panel to display the Display/Update LOCATIONS panel, shown in the following figure.

This panel displays the rows in the LOCATIONS table in the CDB. You can use the following line commands to update the LOCATIONS table:

D   Deletes the row.
I   Inserts a new row. Row values can be entered on the next panel.
U   Updates the row. Row values can be changed on the next panel.

```
ADB2Z5L n ------------ DSN9 Display/Update LOCATIONS ------- Row 1 to 11 of 20
Command ===>                                              Scroll ===> PAGE

                                                 DB2 System: DSN9
Line commands:
 D - Delete  I - Insert  U - Update  DIS - Display location  S - Select
 ALIAS - Aliases for location  LU - LU name  IP - IP name
 ILU - Insert LU  IIP - Insert IP name


                    Link
Select Location     Name     Port      TP Name DBALIAS   TRUSTED SECURE
       *            *        *          *       *         *       *
------ --------------->  ------->  --------->  ------->  --------->  ------- ------
       STLEC1         QMFEC01  446                                N       Y
       DSN8           STM4DSN8 8028                               N       N
       STPLEX4A_DSN7  STM4DSN7 8020                               N       N
       DSN9           STM4DSN9 8016                               N       N
       QMFAIX82       RSNAKE   50002                              N       N
       SQLV73A        VMRACFDB 7300
       SQLV74A        VMRACFDB 7400
```

Figure 391. Display/Update LOCATIONS panel (ADB2Z5L)

Displaying or updating the LUNAMES table

Use the Display/Update LUNAMES panel to update the LUNAMES table.

Select option 1 on the Display/Update communications database (CDB) panel to display the Display/Update LUNAMES panel, as shown in the following figure.

This panel displays the rows in the LUNAMES table in the CDB. You can use the following line commands to update the LUNAMES table:

D   Deletes the row
I   Inserts a new row. Row values can be entered on the next panel.
U   Updates the row. Row values can be changed on the next panel.

```
DB2 Admin ------------- DB2X Display/Update LUNAMES --------------- Row 1 of 2
Command ===>

                                                   DB2 System: DB2X
Line commands:
 D - Delete  I - Insert  U - Update  LOC - Locations  LUM - Lu modes
 USER - User names  MODE - Mode select  ILOC - Insert location
 ILUM - Insert LU modes  IMODE - Insert mode  IUSER - Insert user


               System    Security: Encrypt Mode   User
Select LU Name Mode Name In  Out   Password Select Names Generic
       *       *         *   *     *        *      *     *
------ -------> --------- --- ---   -------- ------ ----- -------
                         V   P     Y        N      O     N
       DKLUDB2W          V   A     N        N      O     N
***************************** END OF DB2 DATA ******************************
```

Figure 392. Display/Update LUNAMES panel (ADB2Z51)

Displaying or updating the IPNAMES table

Use the Display/Update IPNAMES panel to update the IPNAMES table.

Select option 2 on the Display/Update communications database (CDB) panel to display the Display/Update IPNAMES panel, as shown in the following figure.

```
ADB2Z52 ------------- DB2X Display/Update IPNAMES --------------- Row 1 of 1
Command ===>

                                                   DB2 System: DB2X
Line commands:
 D - Delete  I - Insert  U - Update  LOC - Locations  USER - User names
 ILOC - Insert location   IUSER - Insert user


      Link      Security User
Select Name     Out      Names IP address
       *        *         *     *
------ -------> -------- ----- ------------------------------------------------>
       DKIP91   P        O     132.131.61.91
***************************** END OF DB2 DATA ******************************
```

Figure 393. Display/Update IPNAMES panel (ADB2Z52)

Displaying or updating the LUMODES table

Use the Display/Update LUMODES panel to update the update the LUMODES table.

Select option 3 on the Display/Update communications database (CDB) panel to display the Display/Update LUMODES panel, as shown in the following figure.

The Display/Update LUMODES panel displays the rows in the LUMODES table in the CDB. You can use the following line commands to update the LUMODES table:

D   Deletes the row.
I   Inserts a new row. Row values can be entered on the next panel.
U   Updates the row. Row values can be changed on the next panel.

```
DB2 Admin -------------- DB2X Display/Update LUMODES -------------- Row 1 of 1
Command ===>

                                                        DB2 System: DB2X
Line commands:
 D - Delete  I - Insert  U - Update  LU - LU name
                       Conv
Select LU Name  Mode Name Limit
       *        *         *
------ -------> --------- ------
*      DKLUDB2X IBMRDB         5
       STM4DSN6 IBMDSN6M      50
       STM4DSN5 IBMDSN5M      50
***************************** END OF DB2 DATA ******************************
```

Figure 394. Display/Update LUMODES panel (ADB2Z53)

Displaying or updating the MODESELECT table

Use the Display/Update MODESELECT panel to update the MODESELECT table.

Select option 4 on the Display/Update communications database (CDB) panel to display the Display/Update MODESELECT panel, as shown in the following figure.

This panel displays the rows in the MODESELECT table in the CDB. You can use the following line commands to update the MODESELECT table:

D   Deletes the row.
I   Inserts a new row. Row values can be entered on the next panel.
U   Updates the row. Row values can be changed on the next panel.

```
DB2 Admin ----------- DB2X Display/Update MODESELECT ------  ROW 1 TO 21 OF 22
Command ===>

                                                        DB2 System: DB2X
Line commands:
 D - Delete  I - Insert  U - Update  LU  - LU name  LUM - LU modes

Select Auth ID  Plan Name LU Name  Mode Name
       *        *         *        *
------ -------- --------- -------> ------->
                QMF       DKLUDB2X IBMRDRS
                          ST11DB2M IBMDB2LM
                          ST11DB2E IBMDB2LM
                          ST11DB2L IBMDB2LM
                          STM4DSN6 IBMDSN6M
***************************** END OF DB2 DATA ******************************
```

Figure 395. Display/Update MODESELECT panel (ADB2Z54)

Displaying or updating the USERNAMES table

Use the Display/Update USERNAMES panel to update the USERNAMES table.

Select option 5 on the Display/Update communications database (CDB) panel to display the Display/Update USERNAMES panel, as shown in the following figure.

This panel displays the rows in the USERNAMES table in the CDB. You can use the following line commands to update the USERNAMES table:

D   Deletes the row.
I   Inserts a new row. Row values can be entered on the next panel.
U   Updates the row. Row values can be changed on the next panel.

```
 DB2 Admin ----------- DB2X Display/Update USERNAMES --------------- Row 1 of 2
 Command ===>

                                                     DB2 System: DB2X
 Line commands:
  D - Delete  I - Insert  U - Update  LU - LU name  IP  - IP name

 Select T Auth ID  Link      New ID   Password
        * *        *         *        *
 ------ - -------- -------- -------- --------
        O
        O SYSADM   DKLUDB2X NORMUSR
 ***************************** END OF DB2 DATA ******************************
```

Figure 396. Display/Update USERNAMES panel (ADB2Z55)

Displaying or updating the LULIST table

Use the Display/Update LULIST panel to update the LULIST table.

Select option 6 on the Display/Update communications database (CDB) panel to display the Display/Update LULIST panel, as shown in the following figure.

This panel displays the rows in the LULIST table in the CDB. You can use the following line commands to update the LULIST table:

D   Deletes the row.
I   Inserts a new row. Row values can be entered on the next panel.
U   Updates the row. Row values can be changed on the next panel.

```
 DB2 Admin ------------ DB2X Display/Update LULIST -----------------------------
 Command ===>

                                                     DB2 System: DB2X
 Line commands:  D - Delete  I - Insert  U - Update  LU - LU name

        Link      Generic
 Select Name      LU Name
        *         *
 ------ -------> ------>
        DKLUDB21 DKLUDB2
        DKLUDB22 DKLUDB2
 ***************************** END OF DB2 DATA ******************************
```

Figure 397. Display/Update LULIST panel (ADB2Z56)

Displaying DDF

You can display the status and configuration of the distributed data facility (DDF) for your DB2 subsystem.

About this task

You can use the Display DDF panel to display the DDF information for your DB2 subsystem. To display DDF information:

Procedure

1. Select option DF on the System Administration panel. The Display DDF panel is displayed, as shown in the following figure.

```
ADBPZDF n -------------------- DSNB Display DDF ----------------------- 10:28
COMMAND ===>

 -DISPLAY DDF
  ALIAS . . . . . . . . .                    (Name)
  DETAIL  . . . . . . . .                    (Yes/No)
  Output to . . . . . . .                    (T - Table, B - Browse)
```

Figure 398. Display DDF panel (ADBPZDF)

Panel ADBPZDF helps you to construct a DB2 DISPLAY DDF command, which displays the DDF information in a report. You can specify the following options for the -DISPLAY DDF command:

ALIAS
  Displays information specific to the DDF location alias specified by **alias-name**.

DETAIL
  Specifies whether to display additional statistics and configuration information.

Output to
  Specifies where to store the result of the DISPLAY DDF command. Select T (Table) to display the results in an ISPF table, or B (Browse) to display a report.

2. Specify the Alias and Detail fields, then specify one of the following for the Output to field:

   a. Specify **T** in the Output to field to write output to a table. The Display DDF panel is displayed, as shown in the following figure.

```
DB2 Admin ----------------- DB2X Display DDF --------------- Row 1 to 1 of 1
Command ===>                                              Scroll ===> CSR


Sel St   Loc   Luname Genclu  Tpt  Spt  Rpt Ipname  Ipv4   Ipv6   Sql domain
     *    *     *      *        *    *    * *         *      *      *
--- --->  ----- ----->  ------ ---- ---- ---- ------- ----->  ------ ------------>
    STAR DSNA  USIBMS -NONE  8107 8108 8109 -NONE   ::9.30      stplex4a.svl.
****************************** END OF DB2 DATA ******************************
```

The following fields are available on this panel:

SEL
> You can use the / line command to view details for each of the fields on the panel.

St Displays the DDF status

Loc
> Displays the location name of the DDF as it is recorded in the bootstrap data set (BSDS).

Luname
> Displays the DDF LU name as recorded in the BSDS.

Genclu
> Displays the DDF generic LU name as recorded in the BSDS.

Tpt
> Displays the TCP/IP port number for the SQL listener as recorded in the BSDS.

Spt
> Displays the TCP/IP port number for the secure SQL listener as recorded in the BSDS.

Rpt
> Displays the TCP/IP port number for the two-phase commit resynchronization (resync) listener, as recorded in the BSDS.

Ipname
> Displays the IPNAME value as recorded in the BSDS.

Ipv4
> Displays the IP address of the DDF using IPV4 format.

Ipv6
> Displays the IP address of the DDF using IPV6 format.

Sqldomain
> displays the TCP/IP domain name that is associated with the DDF.

Figure 399. Display DDF panel (ADBPZDF)

> b. Specify **B** (Browse) in the `Output to` field to browse the DDF information. The report is displayed, as shown in the following example figure.

```
*************************************** Top of Data ******************
DSNL080I  #DSNA- DSNLTDDF DISPLAY DDF REPORT FOLLOWS:
DSNL081I STATUS=STARTD
DSNL082I LOCATION            LUNAME           GENERICLU
DSNL083I DSNA               USIBMSTM.STM4DSNA -NONE
DSNL084I TCPPORT=8107  SECPORT=8108  RESPORT=8109  IPNAME=-NONE
DSNL085I IPADDR=::9.30.5.16
DSNL086I SQL    DOMAIN=stplex4a.svl.ibm.com
DSNL105I CURRENT DDF OPTIONS ARE:
DSNL106I PKGREL = COMMIT
DSNL099I DSNLTDDF DISPLAY DDF REPORT COMPLETE
***************************************** Bottom of Data ***************
```

Figure 400. Display DDF panel (ADB2DB2O)

Displaying or cancelling distributed threads

You can cancel processing for distributed data facility (DDF) threads that originate locally and access remote data, or that originate remotely and access local data.

About this task

To cancel processing for distributed data facility (DDF) threads that originate locally and access remote data, or that originate remotely and access local data:

Procedure

1. Select option DC on the System Administration panel. The Display/Cancel Distributed Threads panel is displayed, as shown in the following figure.

```
DB2 Admin ------- DB2X Display/Cancel Distributed Threads ---  ROW 1 TO 2 OF 2
Command ===>

Line commands:
 CAN - Cancel thread  DIS - Display thread details

Sel Name     St A   Req ID          Auth ID  Plan     ASID  Luwid
    *        *  *  *    *            *        *        *     *
--- -------- -- -- ----- ------------ -------- -------- ---- ------
    TSO      TR *   255 ISTJE        ISTJE    ADB      008D 2440
     DKIBM000.DKLUDB2X.AB16480C5ADD=2440 ACCESSING DATA AT
     DENMARK_DB2X
    BATCH    TR      3 DB2XDTS       IS512C1  DSNTEP2  008C 2441
     DKIBM000.DKLUDB2X.AB164981904B=2441 ACCESSING DATA AT
     NORDIC_DB2X
***************************** END OF DB2 DATA ****************************
```

Figure 401. Display/Cancel Distributed Threads panel (ADB2ZDC2)

The following fields are available on this panel:

SEL
Input field where you enter one of the line commands listed on the panel.

NAME
Connection name.

ST Connection status.

A Active indicator.

REQ
Number of DB2 requests.

ID Correlation ID.

AUTH ID
> Authorization ID.

PLAN
> Plan name.

ASID
> Address space ID.

LUWID
> Logical unit-of-work ID.

2. Issue one of the following line commands:

   - CAN to cancel a thread. When you press Enter, DB2 Admin issues the CANCEL DDF THREAD command.

   - DIS to display detailed information about a thread. When you press Enter, DB2 Admin issues the DB2 – DISPLAY THREAD DETAILS command.

     The following figure shows the type of information DB2 Admin returns when you issue the DIS line command to display information about a thread.

```
 DB2 Admin ------- DB2X Browse DB2 Command Output ---  Line 00000000 Col 001 080
 Command ===>

 -DIS THD(*) LUWID(2440) DETAIL


 ******************************** TOP OF DATA  *********************************
 DSNV401I < DISPLAY THREAD REPORT FOLLOWS -
 DSNV402I < ACTIVE THREADS -
 NAME     ST A   REQ ID          AUTHID    PLAN    ASID
 TSO      TR *   256 ISTJE       ISTJE     ADB     008D
 -DKIBM000.DKLUDB2X.AB16480C5ADD=2440 ACCESSING DATA AT
 -DENMARK_DB2X
 --LOCATION       SESSID          A ST TIME
 --DENMARK_DB2X   F0839112CD27CFBC   S1 9513816160825
 DISPLAY ACTIVE REPORT COMPLETE
 DSN9022I < DSNVDT '-DIS THD' NORMAL COMPLETION
 ****************************** BOTTOM OF DATA *********************************
```

Figure 402. Display Distributed Threads panel (ADB2DB2O)

The information DB2 Admin returns to you from the commands is in ISPF browse format.

Displaying location details and threads

You can display statistics about threads with a distributed relationship, or display conversation information about DB2 system threads that interact with VTAM.

About this task

To display statistics about threads with a distributed relationship, or display conversation information about DB2 system threads that interact with VTAM:

Procedure

1. Select option DL on the System Administration panel. The Display Active Locations panel is displayed, as shown in the following figure.

```
DB2 Admin ----------- DB2X Display Active Locations ------------- Row 1 of 1
Command ===>                                              Scroll ===> PAGE

Line commands:
 DIS - Display location details  DIST - Display threads

Select Location          PRDID    Linkname         Requesters Servers  Convs
       *                 *        *                 *          *        *
------ ---------------- -------- ---------------- ---------- -------- --------
       DENMARK_DB2P      DSN04010 DKLUDB2P                  0        1        3
       DENMARK_DB2X      DSN05010 DKLUDB2X                  0        0        2
       NORDIC_DB2P       DSN05010 NOLUDB2P                  0        0        2
       NORDIC_DB2R       DSN05010 NOLUDB2R                  0        0        2
       NORDIC_DB2T       DSN05010 NOLUDB2T                  0        0        2
       NORDIC_DB2X       DSN05010 NOLUDB2X                  0        0        2
***************************** END OF DB2 DATA ******************************
```

Figure 403. Display Active Locations panel (ADB2ZDL2)

The following fields are available on this panel:

SELECT
Input field where you enter one of the line commands listed on the panel.

LOCATION
Location name.

PRDID
Database product.

LINKNAME
LU name.

REQUESTERS
Number of requestors.

SERVERS
Number of servers.

CONVS
Number of conversations.

2. Issue one of the following line commands:
   - DIS to display detailed information about a thread. When you press Enter, DB2 Admin issues the DB2 – DISPLAY THREAD DETAILS command.
   - DIST to display the threads. When you press Enter, DB2 Admin issues the DB2 – DISPLAY THREAD command.

   The information DB2 Admin returns to you from the commands is in ISPF browse format.

Starting DDF

You can start DDF.

About this task

To start DDF:

Procedure

Select option DT on the System Administration panel, and press Enter. DB2 Admin issues the DB2 -STA DDF command and displays the status of the command in an

ISPF browse session, as shown in the following figure.

```
DB2 Admin ------- DB2X Browse DB2 Command Output ---  Line 00000000 Col 001 080
Command ===>                                                Scroll ===> PAGE

 -STA DDF


******************************* TOP OF DATA  ********************************
DSNL021I Û START DDF COMMAND ACCEPTED
****************************** BOTTOM OF DATA *******************************
```

Figure 404. Start DDF panel (ADB2DB2O)

Stopping DDF

You can stop the distributed data facility (DDF) if it has already been started.

About this task

To stop the distributed data facility (DDF) if it has already been started:

Procedure

1. Select option DS on the System Administration panel. The Stop DDF panel is
   displayed, as shown in the following figure.

```
DB2 Admin ----------------------- DB2X Stop DDF ----------------------- 16:16
Command ===>

 -STOP DDF

  MODE(
  Stop mode          ===>             (Quiesce or Force, default is quiesce)
)
```

Figure 405. Stop DDF panel (ADB2ZDS)

2. Enter Quiesce or Force in the **Stop Mode** field.
3. Press Enter. DB2 Admin issues the DB2–STOP DDF command. The information
   DB2 Admin returns to you from the command is in ISPF browse format.

Managing stored procedures

You can manage stored procedures.

About this task

To manage stored procedures:

Procedure

1. Select option PM on the System Administration panel. The Manage Stored
   Procedures panel is displayed, as shown in the following figure. This panel lists
   the stored procedure-related operations that are supported by DB2 Admin. The
   format of this panel varies depending on the version of DB2 that you are using.

```
 DB2 Admin --------------- DB2X Manage Stored Procedures --------------- 00:09
 Option ===>

                                                      DB2 System: DB2X
   1 - Display/alter stored procedures               DB2 SQL ID: ISTJE
   2 - Create stored procedure
   3 - Display stored procedure statistics
   4 - Start all stored procedures
   5 - Stop all stored procedures
   6 - Create view on SYSIBM.SYSROUTINES
   7 - Display views on SYSIBM.SYSROUTINES

   Stored procedure catalog table/view for option 1:
     Owner ===>                 (default is SYSIBM)
     Name  ===>                 (default is SYSROUTINES)

 Stored procedures are also available from option 1.0
```

Figure 406. Manage Stored Procedures panel (ADB2ZP)

2. Select an option and press Enter. If you choose option 1, fill in the **Owner** and **Name** fields. When you press Enter, another panel is displayed that corresponds to the option that you chose.

Displaying or altering stored procedures

You can display or alter stored procedures.

About this task

To display or alter stored procedures:

Procedure

Select option 1 on the Manage Stored Procedures panel. TheDisplay/Alter Stored Procedures panel is displayed, as shown in the following figure.
This panel shows the stored procedures you have defined in your system.

```
 DB2 Admin --------------- DB2X Stored Procedures --------------- Row 1 of 11
 Command ===>                                          Scroll ===> PAGE

 Line commands:
 AH - Schema Auth  A - Auth  DROP - Drop  AL - Alter  K - Package  PA - Parms
 DIS - Display  STO - Stop  STA - Start  GR - Grant  COM - Comment
 ? - Show all line commands
                                                   S
                                              Res  Q S P C External
 Sel  Schema   Name             Version A Lang Parms Set O L R T R Name
      *        *                *        * *   *     * * * * * * *
 ---- -------- ---------------- -------> - ---- ------ --- - - - - - -------->
      SMITHJR  PJ                         PLI    0    0 E M N M N PJ
      SMITHJR  PJCOPD2          V1      Y SQL    5    0 N M N   N
      SMITHJR  PJCOPED                    PLI    5    0 E M N M N PJCOPED
      SMITHJR  PJJAVAPRC                  JAVA   0   10 E M N S N PKG402110
      SMITHJR  PJNSP            DISABLED N SQL   1    0 N M N   N
      SMITHJR  PJNSP            VER1     Y SQL   1    0 N M N   N
      SMITHJR  PJNSP            VER2     N SQL   1    0 N M N   N
      SMITHJR  PJNSP            VER3     N SQL   1    0 N M N   N
      SMITHJR  PJNSP            VER4     N SQL   1    0 N M N   N
 ***************************** END OF DB2 DATA *****************************
```

Figure 407. Display/Alter Stored Procedures panel (ADB21O)

The following fields are available on this panel:

SEL
    Input field where you enter one of the line commands listed on the panel.

SCHEMA
Schema of the stored procedure.

NAME
Name of the stored procedure.

VERSION
Version of the native SQL procedure.

A Active. Identifies the active version of a native SQL procedure.

LANGUAGE
Implementation language.

PARMS
Number of parameters for the stored procedure.

LANGUAGE
Implementation language.

RES SET
Maximum number of result sets that can be returned.

O Origin of the native SQL procedure.

SQL
Whether SQL statements are allowed, which is one of the following:
N   Contains no SQL statements
C   Contains SQL statements
R   Reads SQL data
M   Modifies SQL data

SR Whether the program should remain resident when it ends.
Y   Program remains resident
N   Program does not remain resident
blank Not external or user-defined function.

PT Program type, which is one of the following:
M   Main
S   Subroutine

CR Commit on return.
Y   Program is committed immediately.
N   Program continues.
A   Autonomous. Only the unit of work from the procedure is committed. Work from the application that calls the procedure is not immediately committed.

EXTERNAL NAME
Load module name for the stored procedure.

Note: The SRC line command is not supported for native SQL procedures. Press PF1 if you get an invalid line command message and look at the **O** column. If there is an **N** in that column, then the SRC command is not supported. An E in the O column indicates the SRC command is supported.

Creating stored procedures

You can create a stored procedure.

About this task

Restriction: When creating SQL stored procedures, the maximum length of the procedure body is 2MB (32,767KB).

To create a stored procedure:

Procedure

1. Select option 2 on the Manage Stored Procedures panel. The Create Stored Procedure panel is displayed, as shown in the following figure.

```
DB2 Admin ------------------- DB2X Create Procedure ------------------- 11:00
Command ===>

 CREATE PROCEDURE

Schema . . . . . SMITHJR  >     (Default is SMITHJR)
Name . . . . . . PJNEWSP         > (? to look up existing procedures)

(
Number of parameters . . 1     (0-255)
)

LANGUAGE . . . . SQL            (ASSEMBLE,C,PLI,COBOL,REXX,JAVA,SQL)

Native SP  . . . Y              (Yes,No)
  VERSION  . . . V7             > (optional, default is V1)
```

Figure 408. Create Stored Procedure panel (ADB26CO)

2. Enter the required parameters and press Enter to continue with the create operation, or press End to avoid creating a procedure.
3. On the Create Stored Procedure Parameters panel, enter the stored procedure parameters for the language you specified. For example, the language SQL procedure types panel is shown in the following figure. DB2 Admin issues the SQL CREATE PROCEDURE statement with the parameters you specify.

```
DB2 Admin ------------------- DB2X Create Stored Procedure Parameters------------- 11:00
Command ===>

CREATE PROCEDURE "PJ_NP" ..
(parameter number 1)      LANGUAGE SQL ..

Parm type  . . . IN                  (IN, OUT, or INOUT)
Parm name  . . . P1            >      (Parameter name)

For a non table like parameter specify:
Schema . . . . .             >       (For user-defined type)
Data type  . . . BIGINT          >   (Built-in or ? to look up user types
Length . . . . .                     (1 if DBCLOB with units indicator G)
Scale  . . . . .

FOR ? DATA . . .                     (BIT, SBCS, or MIXED)
CCSID  . . . . .                     (ASCII, EBCDIC, or UNICODE)
AS LOCATOR . . .                     (Yes/No)

For a TABLE LIKE parameter specify:
  Table owner  . .          >        (Default is SMITHJR)
  Table name . . .               >   (Table parameter, ? to look up)
```

Figure 409. Create Stored Procedure Parameters panel (ADB26COU)

Creating native SQL procedures

You can use the CREATE SQL procedure to create a native SQL procedure.

About this task

Restriction: The maximum length of the native SQL procedure body is 2 MB
(32,767 KB).

You can create a native SQL procedure to help you with commonly performed
tasks. For example, if you often need to create a test database, you can create a
native SQL procedure to create a test database every time that you need to do so.
You can also use other functions within DB2 Admin to generate the native SQL
procedure's DDL and to reuse that DDL for a different database and its objects.

To create a native SQL procedure that creates a test database:

Procedure

1. Select option **2.4** on the Administration Menu and then enter option **CO**. The
   Create Procedure panel is displayed.

```
 DB2 Admin -------------------- DB2X Create Procedure -------------------- 11:00
 Command ===>

  CREATE PROCEDURE

 Schema . . . . .           >      (Default is SMITHJR)
 Name . . . . . . SPTDEMO1           >  (? to look up existing procedures)

 (
 Number of parameters . . 0      (0-255)
 )

 LANGUAGE . . . . SQL              (ASSEMBLE,C,PLI,COBOL,REXX,JAVA,SQL)

 Native SP . . . YES              (Yes,No)
   VERSION  . . . V1                >  (optional, default is V1)
```

Figure 410. Create Procedure panel (ADB26CO)

2. Enter the required parameters and press Enter.
3. On the Create SQL Procedure Body panel, enter the SQL procedure body. For
   example, enter CREATE DATABASE DBDEMO1, as shown in the following
   figure:

```
ADB26COQ  ----- DB2X Create SQL Stored Procedure Body ---- Columns 00001 00072
 Command ===>                                               Scroll ===> CSR

 CREATE PROCEDURE "SPTDEMO1" ..
 ****** **************************** Top of Data *****************************
 ==MSG> -Warning- The UNDO command is not available until you change
 ==MSG>          your edit profile using the command RECOVERY ON.
 ''''''
 ''''''   CREATE DATABASE DBDEMO1
 ''''''
 ''''''
 ''''''
 ''''''
 ''''''
 ''''''
 ''''''
 ''''''
 ''''''
 ''''''
 ''''''
 ''''''
 ****** **************************** Bottom of Data ***************************
```

Figure 411. Create Stored Procedure Parameters panel (ADB26COQ)

> Now that the native SQL procedure has been created, use either the DDL line
> command or the GEN function to generate the DDL. The next steps show you
> how to use the GEN function to generate the DDL with masking into a work
> statement list.

4. Select **option 1 - DB2 system catalog** on the Administration Menu. Then select
   option O on the System Catalog panel to display the Stored Procedures panel
   ADB210. Type GEN next to your SQL procedure name.

```
ADB210 in --------------- DB2X Stored Procedures ------------ Row 1 to 1 of 1
 Command ===>                                               Scroll ===> CSR

 Commands: GRANT
 Line commands:
  AH - Schema Auth  A - Auth  DROP - Drop  AL - Alter  K - Package  PA - Parms
  DIS - Display  STO - Stop  STA - Start  GR - Grant  COM - Comment
  ? - Show all line commands
                                                       S
                                                  Res   Q S P C External
 Sel  Schema    Name              Version  A Lang Parms  Set O L R T R Name
      *         *                 *        * *          *   * * * * * *
 ---- -------- ------------------ -------- - ---- ------ --- - - - - - ---------
 GEN  SYSADM   SPTDEMO1           V1       Y SQL      0   0 N M N   N
 ***************************** END OF DB2 DATA ******************************
```

Figure 412. Stored Procedures panel (ADB210)

> The Generate SQL from DB2 catalog panel (ADB2GENS) is displayed.

5. Use the GEN function to generate the DDL with masking into a work
   statement list. Specify YES in the **CREATE PROCEDURE**, **Use Masking**, and
   Add to work stmt list list fields.

```
ADB2GENS  -------------- DB2X Generate SQL from DB2 catalog ------------- 15:36
 Option ===>

  Generate SQL statements for:                           DB2 System: V91A
    stored procedure SYSADM.SPTDEMO1                      DB2 SQL ID: SYSADM
                                                                    More:      +
  SQL statement types to be generated from the DB2 catalog:

    CREATE PROCEDURE . . . . : Y (Y,N,A) GRANT access ON PROCEDURE. : N (Y,N)
    REBIND PLAN. . . . . . . : Y (Y,N,D) COMMENT ON . . . . . . . . : Y (Y,N)

  New names/values for generated SQL: (leave blank to use current values)
    Object owner. . . . . :          >  Run SQLID. . . . . . . . . :
    Object grantor. . . . :
    Object schema . . . . :          >
    Target DB2 version. . :             (Current DB2 version: 915)
    Use Masking . . . . . : YES         (Yes/No)
    Use Exclude Specification YES       (Yes/No)
    Generate catalog stats: NO          (Yes/No/Only)
      Target cat qualifier:          >  (Default is SYSIBM)
      Statistics tables . . S           (All or Select. Defalut is All)
  Output file and execution mode:
    Add to work stmt list : YES         (Yes/No)
    Data set name . . . . : 'SYSADM.NSPDMO1.DDL
 Data set disposition. OLD           (OLD, SHR, or MOD)
  Execution mode. . . . . BATCH       (BATCH or TSO)
  Commit statements per .             (All or None. Default is All)
  DB2 defaults handling .             (Keep or Remove. Default is Keep)

 DB2 Command output file:
   Data set name . . . . .
     Data set disposition. OLD         (OLD, SHR, or MOD)

 BP - Change batch job parameters
```

Figure 413. Generate SQL from DB2 catalog panel (ADB2GENS)

The ADB2EDIT panel is displayed.

6. On the ADB2EDIT panel, specify masks for the SQL stored procedure.. In the
   following example, specify SPTDEMO1, SPTDEM02 for STPNAME and DBDEM01,
   DBDEM02 for DBNAME.

```
ADB2EDIT  --------------------  -------------------------- Columns 00001 00072
 Command ===>                                              Scroll ===> CSR
==MSG>     - To support/migrate DB2V8 masking input,OWNER,TBOWNER and
==MSG>       IXOWNER will mask both owner and schema fields.SCHEMA,
==MSG>       TBSCHEMA and IXSCHEMA will be applied to schema fields only.
==MSG>     - SINGLECH format is SINGLECH:<character>[,<escape character>]
==MSG>       where the single character in a mask specification represents
==MSG>       any character at that position. If the specified escape character
==MSG>       precedes the specified single character, then the single character
==MSG>       is treated as a literal.
==MSG>     - The view, alias, and synonym mask (both name and schema) apply only
==MSG>       to the CREATE statement for these objects. For example, VWNAME is
==MSG>       valid only for the CREATE VIEW vwname statement. All other usages
==MSG>       of these names and schemas are vague and can also refer to table
==MSG>       names and schemas. These other usages can be masked only by TBNAME
==MSG>       if the view names are being changed for both the CREATE statement and
==MSG>       SQL that use this view.
==MSG>     - The following masks can not have the object-specific qualifiers
==MSG>       listed in the mask syntax:
==MSG>         NAME, SCHEMA, SETPATHSC, DBNAME, COLLNAME, SFNAME, GRANTID,
==MSG>         GRANTOR, GRANTEE, ROLE, DBROLE, TSROLE, TBROLE, IXROLE,
==MSG>         GBPNAME, TCNAME, XMLSCHID, AUTHID, SQLID, SGNAME, OWNER, BPNAME,
==MSG>         PLNNAME and SINGLECH.
==MSG>
==MSG>   Mask examples:
==MSG>     OWNER:ABC*,DEF*
==MSG>     NAME:PRE*,NPRE*
==MSG>     XMLSCHID:PO1,PO2
==MSG>     WLMENV:WLM33,WLM44
==MSG>     LOCATION:LOC3*,LOCT*
==MSG>     SETPATHSC:SYSIBM,SYSFUN
==MSG>     SINGLECH:_
==MSG>     SINGLECH:_,+
==MSG>
==MSG>   Object-specific mask examples:
==MSG>     TBSCHEMA:CREATOR1.TB2:CREATOR1,NEW_CRE1
==MSG>     IXNAME:IXOWN*.IX3*:IX3*,IX4*
==MSG>     IXBPNAME:IXOWN1.INDX2:BP1,BP3
==MSG>
==MSG>   Overwrite examples:
==MSG>     COMPRESS:MYDB*.MYTS*,YES
==MSG>     SEGSIZE:MYDB*.MYTS*,8
==MSG>     DSSIZE:MYDB*.MYTS*,4G
==MSG>     PRIQTY:*.*,REXX(MYPRIQTY,DBNAME='MYDBTEST')
==MSG>     TSPRIQTY:MYDB*.MYTS*,30
==MSG>     IXPRIQTY:MYCR*.MYIX*,25%
==MSG>     IXSECQTY:MYCR*.MYIX*,REXX(MYSECQTY,IXNAME,IXCREATOR,PCT=20%)
==MSG>     DEFER:USER001.*IXNAME,NO
==MSG>     DEFINE:DBNAME*.*TSPC,REXX(MYDEFINE,DEFINE='YES')
==MSG>     HASHSPC:TBCREATOR.MYTBNAME,100M
==MSG>     TBINLOBL:TBCREATOR.MYTBNAME.COLNAME,16000
==MSG>     DTINLOBL:DTCRE*.DTNAME*,16000
==MSG>     IXCLOSE:MYCR*.MYIX*,NO
==MSG>     AUDIT:MYDB*.MYTB*,CHANGES
==MSG>     TRACKMOD:MYDB*.MYTS*,NO
==MSG>     DCAPTURE:TBCRE*.MYTB*,NONE
==MSG>
==MSG>   Syntax for info about renamed objects/columns:
==MSG>     renameobj:old-name,new-name
==MSG>     RENAMECOL:table-name.old-colname,new-colname
==MSG>       ( + in col 72 indicates continuation on next line col 1)
==MSG>   renameobj:
==MSG>     RENAMEDB, RENAMETS, RENAMETB, RENAMEIX,
==MSG>     RENAMEGV
==MSG>   Examples:
==MSG>     RENAMETB:OLDOWNER.OLDNAME,NEWOWNER.NEWNAME
==MSG>     RENAMECOL:OWNERA.MYTB.OLDCOLNAME,NEWCOLNAME
==MSG>
 000100 STPNAME:SPTDEMO1,SPTDEMO2
 000200 DBNAME:DBDEMO1,DBDEMO2
 ****** **************************** Bottom of Data ****************************
```

Figure 414. ADB2EDIT panel

7. Return to panel ADB2GENS and specify the work statement list data set name and the work statement list name.

```
ADB2GENS  -------------- DB2X Generate SQL from DB2 catalog ------------- 15:36
 Option ===>

  Generate SQL statements for:                         DB2 System: V91A
    stored procedure SYSADM.SPTDEMO1                    DB2 SQL ID: SYSADM
                                                                   More:     +
  SQL statement types to be generated from the DB2 catalog:
  EssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssN
  e ADB2WLDA  ----------------- Specify Work Statement List ---------------- e
  e                                                                          e
  e                                                                          e
  e  Work stmt list dsn  . . . 'SYSADM.NSPDEMO2.WSL'                         e
  e  Work stmt list name . . . NSPDEMO2                                      e
  e                                                                          e
  e                                                                          e
  e                                                                          e
  e                                                                          e
  e                                                                          e
  e                                                                          e
  e                                                                          e
  e                                                                          e
  DssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssM
    Data set name . . . . : 'SYSADM.NSPDMO1.DDL
```

Figure 415. Generate SQL from DB2 catalog panel (ADB2GENS)

8. Open the DB2 Admin menu and select the **W** option for WSL. The ADB2W panel is displayed. Select option 1 to show the work statement list library.

```
ADB2W min --------------- DB2X Manage Work Statement Lists -------------- 15:43
 Option ===> 1

   1 - Show work statement list library       DB2 System: V91A
   2 - Show work statement list               DB2 SQL ID: SYSADM

  Work stmt list dsn  ===> 'SYSADM.NSPDEMO2.WSL'
  Work stmt list name ===> NSPDEMO2
```

Figure 416. Manage Work Statement Lists panel (ADB2W)

9. On the ADB2W1 panel enter the S line command to show the work statement list.

```
ADB2W1 in ------------ Work Statement List Library: 'SYSADM.NSP Row 1 to 1 of 1
 Command ===>                                            Scroll ===> CSR

 Line commands:
  S - Show  R - Run in batch  D - Delete  C - Copy  A - Append  Q - Clone
  I - Interpret  V - Validate  E - Edit  O - Run online

 Sel Name     Created    Changed          ID
     *        *          *                *
 --- -------- ---------- ---------------- --------
 S   NSPDEMO2 2009/06/01 2009/06/01 15:43 SYSADM
 ****************************** END OF DB2 DATA *******************************
```

Figure 417. Work Statement List Library panel (ADB2W1)

The work statement list is displayed:

```
ADB2W1S n ----------- Show Work Statement List: NSPDEMO2 -- Row 27 to 39 of 39
 Command ===>                                           Scroll ===> CSR

 Line commands:
  D - Delete  I - Insert  E - Edit  C - Copy  M - Move  A - After  B - Before
  R - Repeat

 Select Type Statement
        *    *
 ------ ---- -------------------------------------------------------------------->
        COM  --
        DML   SET CURRENT PATH = "SYSIBM","SYSFUN","SYSPROC","SYSADM"
        COM  --
        DDL   CREATE PROCEDURE SYSADM.SPTDEMO2..   ()..   VERSION V1..   LAN
        COM  --
        DML   COMMIT
        COM  --
        COM  --#SET TERMINATOR ;
        COM  --
        COM  ----------------------------------------------------------------
        COM  -- ADB2GEN - End of generated DDL
        COM  ----------------------------------------------------------------
        COM  --
```

Figure 418. Show Work Statement List panel

10. Return to panel ADBW1 and enter the V line command to validate the work
    statement list.

```
ADB2W1 in ----------- Work Statement List Library: 'SYSADM.NSP Row 1 to 1 of 1
 Command ===>                                           Scroll ===> CSR

 Line commands:
  S - Show  R - Run in batch  D - Delete  C - Copy  A - Append  Q - Clone
  I - Interpret  V - Validate  E - Edit  O - Run online

 Sel Name     Created    Changed          ID
     *        *          *                *
 --- -------- ---------- ---------------- --------
 V   NSPDEMO2 2009/06/01 2009/06/01 15:43 SYSADM
 ****************************** END OF DB2 DATA ********************************
```

Figure 419. Work Statement List Library panel

The Validation Work Statement List Report is displayed:

```
SDSF OUTPUT DISPLAY VLDNSP2  JOB00083  DSID   105 LINE 27       COLUMNS 02- 81
 COMMAND INPUT ===>                                             SCROLL ===> CSR
                           RENAME statements.
--------------------------------------------------------------------------------


VALIDATE WORK STATEMENT LIST REPORT
===================================

Prepared on V91A (DB2 Release 915) by SYSADM at 2009-06-01 16:10
for SYSADM.NSPDEMO2.WSL(NSPDEMO2)

ADB3020W Warning for Procedure SYSADM.SPTDEMO2M in CREATE/ALTER Procedure NSP body
statement:
 Objects referenced in
         Create/Alter/Comment/Drop/Exchange/Label/Rename may or may not exist during NSP
runtime

CREATED OBJECTS
---------------
  Procedure SYSADM.SPTDEMO2M
```

Figure 420. Validation Work Statement List Report

11. After you validate the work statement list, enter the R line command to run
    the JCL job.

```
ADB2W1 in ----------- Work Statement List Library: 'SYSADM.NSP Row 1 to 1 of 1
 Command ===>                                               Scroll ===> CSR

 Line commands:
  S - Show  R - Run in batch  D - Delete  C - Copy  A - Append  Q - Clone
  I - Interpret  V - Validate  E - Edit  O - Run online

 Sel Name     Created    Changed          ID
     *        *          *                *
 --- -------- ---------- ---------------- --------
 R   NSPDEMO2 2009/06/01 2009/06/01 16:02 SYSADM
 ****************************** END OF DB2 DATA ******************************
```

Figure 421. Work Statement List Library panel (ADB2W1)

12. Return to panel ADB210 and verify that the SPTDEMO2 native SQL procedure
    was created successfully.

```
ADB210 in ---------------- DB2X Stored Procedures ------------ Row 1 to 2 of 2
 Command ===>                                               Scroll ===> CSR

 Commands: GRANT
 Line commands:
  AH - Schema Auth  A - Auth  DROP - Drop  AL - Alter  K - Package  PA - Parms
  DIS - Display  STO - Stop  STA - Start  GR - Grant  COM - Comment
  ? - Show all line commands
                                                    S
                                              Res   Q S P C External
 Sel  Schema   Name               Version A Lang Parms  Set O L R T R Name
      *        *                  *       * *        *    * * * * * * *
 ---- -------- ------------------ -------- - ---- ------ --- - - - - - ---------
      SYSADM   SPTDEMO1           V1      Y SQL      0    0 N M N  N
      SYSADM   SPTDEMO2           V1      Y SQL      0    0 N M N  N
 ****************************** END OF DB2 DATA ******************************
```

Figure 422. Stored Procedures panel (ADB210)

Results

- You used the CREATE Stored Procedure function to create a native SQL
  procedure.

- You used the GEN function to generate the DDL with masking into a work statement list to create another native SQL procedure.
- You validated and ran the generated work statement list to successfully create the new native SQL stored procedure.

The terminator for each generated statement was ? (question mark) for releases earlier than DB2 Admin Version 11.1 and is the ` (grave accent) for DB2 Admin Version 11.1 and later releases.

Displaying stored procedure statistics

You can display stored procedure statistics.

About this task

To display stored procedure statistics:

Procedure

Select option 3 on the Manage Stored Procedures panel to display the Display Stored Procedure Statistics panel, as shown in the following figure. This panel shows statistics for stored procedures that are accessed by DB2 applications.

```
DB2 Admin ------- DB2X Browse DB2 Command Output ---  Line 00000000 Col 001 080
Command ===>                                            Scroll ===> PAGE

-DIS PROC(SYSPROC.DSNWZP)


******************************** Top of Data **********************************
DSNX940I ? DSNX9DIS DISPLAY PROCEDURE REPORT FOLLOWS -
------- SCHEMA=SYSPROC
DSNX9DIS PROCEDURE DSNWZP HAS NOT BEEN ACCESSED OR IS NOT DEFINED
DSNX9DIS DISPLAY PROCEDURE REPORT COMPLETE
DSN9022I ? DSNX9COM '-DISPLAY PROC' NORMAL COMPLETION
******************************** Bottom of Data *******************************
```

Figure 423. Display Stored Procedure Statistics panel (ADB2DB2O)

Starting all stored procedures

You can start all stored procedures.

About this task

To start all stored procedures:

Procedure

Select option 4 on the Manage Stored Procedures panel. DB2 Admin issues the DB2 START STORED PROCEDURE(*.*) command, and displays the status of the command in an ISPF edit session, as shown in the following figure.

```
DB2 Admin ------- DB2X Browse DB2 Command Output ---  Line 00000000 Col 001 080
Command ===>                                              Scroll ===> PAGE

-STA PROC(*.*)


******************************* Top of Data *********************************
DSNX946I ? DSNX9ST2 START PROCEDURE SUCCESSFUL FOR *.*
DSN9022I ? DSNX9COM '-START PROC' NORMAL COMPLETION
****************************** Bottom of Data *******************************
```

Figure 424. Start All Stored Procedures panel (ADB2DB2O)

Stopping all stored procedures

You can stop all stored procedures.

About this task

To stop all stored procedures:

Procedure

Select option 5 on the Manage Stored Procedures panel. When you press Enter,
DB2 Admin issues the DB2 STOP PROCEDURES(*.*) command and displays the
status of the command in an ISPF edit session, as shown in the following figure.

```
DB2 Admin ------- DB2X Browse DB2 Command Output ---  Line 00000000 Col 001 080
Command ===>                                              Scroll ===> PAGE

-STO PROC(*.*)


******************************* Top of Data *********************************
DSNX947I ? DSNX9SP2 STOP PROCEDURE SUCCESSFUL FOR *.*
DSN9022I ? DSNX9COM '-STOP PROC' NORMAL COMPLETION
****************************** Bottom of Data *******************************
```

Figure 425. Stop All Stored Procedures panel (ADB2DB2O)

Creating views of stored procedures

You can create a view of stored procedures on SYSIBM.SYSROUTINES which is
useful if you want to let people administer their own stored procedures.

About this task

To create a view of stored procedures:

Procedure

1. Select option 6 on the Manage Stored Procedures panel. The Create View on
   SYSIBM.SYSROUTINES panel is displayed, as shown in the following figure.
   This panel lets you define a view for all procedures with the (LIKE) pattern
   you define.

```
DB2 Admin ------------ DB2X Create View on SYSIBM.SYSROUTINES ---------- 00:12
Command ===>

 CREATE VIEW

 Owner    ===> ISTJE     >
 Name     ===> ADB_ROUTINES     >

 AS SELECT *
   FROM SYSIBM.SYSROUTINES
   WHERE SCHEMA LIKE '
 Pattern  ===> ADB%      >  '

 WITH CHECK OPTION ;

 GRANT SELECT,INSERT,UPDATE,DELETE ON (above table) TO
 Grantees ===>
```

Figure 426. Create View on SYSIBM.SYSROUTINES panel (ADB2ZP6)

2. Fill in the fields on this panel to create a view, for example, define view ABC.PROCEDURES as a view on SYSIBM.SYSROUTINES WHERE SCHEMA LIKE 'ABC. View ABC.PROCEDURES contains all stored procedures with the schema starting with ABC. In addition, you can issue GRANT SELECT, INSERT, UPDATE, or DELETE statements on the view to a list of authorization IDs (grantees).

Displaying views of stored procedures

You can display views of stored procedures.

About this task

To display the views that exist on SYSIBM.SYSROUTINES:

Procedure

Select option 7 on the Manage Stored Procedures panel. The Tables, Views, and Aliases panel is displayed, as shown in the following figure. This panel shows the views that exist on SYSIBM.SYSROUTINES; for example, it would show the views created using option 6 on the Manage Stored Procedures panel.

```
DB2 Admin ----------- DB2X Tables, Views, and Aliases ----------- ------

Commands: GRANT      ALL
Line commands:
 C - Columns  A - Auth  L - List  X - Indexes  S - Table space  D - Database
 V - Views  T - Tables  P - Plans  Y - Synonyms  SEL - Select prototyping
 ? - Show all line commands

Sel  Name              Schema   T DB Name  TS Name   Cols      Rows Chks C

----- ------------------ -------- - -------- -------- ------ ----------- ---- -
      PROCEDURES        ISTJE    V DSNDB06  SYSOBJ     79        -1  0
      FUNCTIONS         ISTJE    V DSNDB06  SYSOBJ     79        -1  0
```

Figure 427. Tables, Views, and Aliases panel showing views on SYSIBM.SUBROUTINES (ADB21T)

Managing functions

You can use DB2 Admin to manage functions.

About this task

To manage functions:

Procedure

1. Select option FM on the System Administration panel. The Manage Functions panel is displayed, as shown in the following figure. This panel lists the Functions-related operations that are supported by DB2 Admin.

```
 DB2 Admin ---------------- DB2X Manage Functions ----------------------- 18:35
 Option ===>

                                                       DB2 System: DB2X
   1 - Display/alter functions                         DB2 SQL ID: ISXSTL
   2 - Create functions
   3 - Display function statistics
   4 - Start all functions
   5 - Stop all functions
   6 - Create view on SYSIBM.SYSROUTINES
   7 - Display views on SYSIBM.SYSROUTINES

   Catalog table/view for options 1-2:
     Owner ===> SYSIBM          (default is SYSIBM)
     Name  ===> SYSROUTINES     (default is SYSROUTINES)

 User defined functions can also be managed from option 1.F
```

Figure 428. Manage Functions panel (ADB2ZF)

2. Select an option and press Enter. If you choose option 1, fill in the **Owner** and **Name** fields. When you press Enter, another panel is displayed that corresponds to the option that you chose.

Displaying or altering functions

You can display or alter functions.

About this task

To display or alter functions:

Procedure

Select option 1 on the Manage Functions panel. The Display or Alter Functions panel is displayed, as shown in the following figure.

```
 DB2 Admin -------------------- DB2X Functions -------------------- Row 1 of 44
 Command ===>                                              Scroll ===> PAGE

 Line commands:
  AH - Schema auth  A - Auth  DROP - Drop  AL - Alter   K - Package  PA - Parms
  RT - Return type  DIS - Display  STO - Stop  STA - Start  GEN - Generate DDL
  COM - Comment  I - Interpret  CRE - Create func  GR - Grant  JAR - JAVA detail
  DDL - Object DDL
                                                  D         S
                                           F      E E C P   Q S P E External
 S    Schema   Name    Specific Name      O T Parms T A F S F L R T S Name
      *        *       *                  * * *    * * * * * * * * * *
 ---- -------> -------> ------------------> - - ------ - - - - - - - - - --------
      ISTJE    +        SQL990208100338896 U S    2    N
      ISTJE    -        KR_MINUS           U S    2    N
      ISTJE    BLOB     SQL99020816075424# S S    1    Y
      ISTJE    CHAR     SQL990208160600039 S S    1    Y
      ISTJE    CLOB     SQL99020816074873# S S    1    Y
      ISTJE    D        SQL99020817171170M S S    1    Y
      ISTJE    DATE     SQL99020816083184# S S    1    Y
      ISTJE    DECIMAL  SQL99011815223541B S S    1    Y
      ISTJE    DECIMAL  SQL99021816281595J S S    1    Y
      ISTJE    DECIMAL  SQL99020817171173M S S    1    Y
```

Figure 429. Manage Functions panel (ADB21F)

The Display or Alter Functions panel displays information about all the
user-defined functions in your DB2 subsystem.
The following fields are available on this panel:

S   Input field where you enter one of the line commands listed on the panel.

SCHEMA
     Schema of the function.

NAME
     Name of the function.

SPECIFIC NAME
     Specific name of the function.

O   Origin of the function, which is one of the following:
     E   External
     U   Sourced
     S   System generated
     Q   SQL

FT   Function type, which is one of the following:
     C   Column
     S   Scaler
     T   Table

PARMS
     Number of parameters for the function.

DET
     Whether the external function returns the same result when called using the
     same parameters. This field contains one of the following:
     Y   Yes
     N   No
     blank   The routine is a function, but not an external function.

EA   Whether the external function changes the state of an object that DB2 does not
     manage. This field contains one of the following:
     Y   Yes
     N   No

blank The routine is not an external function.

CF Cast function, which is one of the following:
- **Y** Yes
- **N** No

PS Parameter style, which is one of the following:
- **D** DB2SQL
- **G** General
- **N** General with nulls
- **J** Java™
- **blank** Not external or user-defined function.

F Fenced (applies if it is run separately from DB2).

SQL

Whether SQL statements are allowed, which is one of the following:
- **N** Contains no SQL statements
- **C** Contains SQL statements
- **R** Reads SQL data
- **M** Modifies SQL data
- **blank** Not applicable

SR Whether the program should remain resident when it ends.
- **Y** Program remains resident
- **N** Program does not remain resident
- **blank** Not external or user-defined function.

PT Program type, which is one of the following:
- **M** Main
- **S** Subroutine
- **blank** Not external or user-defined function.

ES External security, which is one of the following:
- **D** DB2 address space user
- **U** User
- **C** Definer
- **blank** Not external or user-defined function.

EXTERNAL NAME

Load module name for the stored procedure. This field is blank if it is not an external or user-defined function.

Creating functions

You can create new, user-defined functions.

About this task

To create a new user-defined function:

Procedure

1. Select option 2 on the Manage Functions panel. The Create Function panel is displayed, as shown in the following figure.

```
DB2 Admin -------------------- DB2X Create Function -------------------- 18:38
Command ===>

 CREATE FUNCTION

 Schema      ===>          >           (Default is ISTJE)
 Name        ===>                      > (? to look up existing functions)

 (
 Number of parameters ===>             (0-255)
 )

 SPECIFIC    ===>                      > (Specific name)
                                                         (continued...)
```

Figure 430. Create Function panel (ADB26CF)

2. Enter the required parameters and press Enter to continue with the create operation, or press End to avoid creating a function. DB2 Admin issues the SQL CREATE FUNCTION statement with the parameters you specify.

   To create a new SQL scalar function:

   Restriction: When creating SQL scalar functions, the maximum length of the return statement is 2MB (32,767KB).

   a. Write the SQL scalar function as part of the CREATE statement.
   b. Pre-compile, compile, and link the program.
   c. If the program has SQL statements, bind a package.
   d. Create the function to register it to DB2 and grant execute to authorize all appropriate users.
   e. Use the function in application programs.

Displaying function statistics

You can display function statistics.

About this task

To display function statistics:

Procedure

Select option 3 on the Manage Functions panel. The Display Function Statistics panel, as shown in the following figure, is displayed. This panel displays statistics about external user-defined functions accessed by DB2 applications.

```
DB2 Admin ------- DB2X Browse DB2 Command Output ---  Line 00000000 Col 001 080
Command ===>                                              Scroll ===> PAGE

 -DIS FUNCTION SPEC(*.*)


******************************** Top of Data *********************************
DSNX975I DB2X DSNX9DIS DISPLAY FUNCTION SPECIFIC REPORT FOLLOWS -
FUNCTION          STATUS ACTIVE QUEUED MAXQUE TIMEOUT   WLM_ENV
APPL1             STARTED    1      0      0        0   PAYROLL
APPL2             STARTED    1      0      0        0   PAYROLL
APPL3             STARTED    0      1      2        0   PAYROLL
APPL5             STOPREJ    0      0      0        0   SANDBOX
APPL6             STOPABN    0      0      0        0   PAYROLL
FUNC1             STOPQUE    0      0      0        0   SANDBOX
DSNX9DIS DISPLAY FUNCTION SPECIFIC REPORT COMPLETE
DSNX975I - DSNX9DIS DISPLAY FUNCTION SPECIFIC REPORT FOLLOWS -
***************************** Bottom of Data *********************************
```

Figure 431. Display Function Statistics panel (ADB2DB2O)

When you press Enter, DB2 Admin issues the -DIS FUNCTION SPEC(*.*)
command.

Starting all functions

You can start all functions.

About this task

To start all functions:

Procedure

Select option 4 on the Manage Functions panel. DB2 Admin issues the -STA
FUNCTION SPEC(*.*) command, and displays the status of the command in an
ISPF edit session, as shown in the following figure.

```
DB2 Admin ------- DB2X Browse DB2 Command Output ---  Line 00000000 Col 001 080
Command ===>                                              Scroll ===> PAGE

 -STA FUNCTION SPEC(*.*)


******************************** Top of Data *********************************
DSNX973I DB2X DSNX9ST2 START FUNCTION SPECIFIC SUCCESSFUL FOR *.*
DSN9022I DB2X DSNX9COM '-START FUNC' NORMAL COMPLETION
***************************** Bottom of Data *********************************
```

Figure 432. Start All Functions panel (ADB2DB2O)

Stopping all functions

You can stop all functions.

About this task

To stop all functions:

Procedure

Select option 5 on the Manage Functions panel. DB2 Admin issues the -STO
FUNCTION SPEC(*.*) command and displays the status of the command in an

ISPF edit session, as shown in the following figure.

```
 DB2 Admin ------- DB2X Browse DB2 Command Output ---  Line 00000000 Col 001 080
 Command ===>                                                  Scroll ===> PAGE

 -STO FUNCTION SPEC(*.*)


******************************** Top of Data **********************************
DSNX974I DB2X DSNX9SP2 STOP FUNCTION SPECIFIC SUCCESSFUL FOR *.*
DSN9022I DB2X DSNX9COM '-STOP FUNC' NORMAL COMPLETION
***************************** Bottom of Data **********************************
```

Figure 433. Stop All Functions panel (ADB2DB2O)

Creating views of functions

You can create a view of a function on SYSIBM.SYSROUTINES which is useful if
you want to let people administer their own functions.

About this task

To create a view of a function:

Procedure

1. Select option 6 on the Manage Functions panel. The Create View on
   SYSIBM.SYSROUTINES panel is displayed, as shown in the following figure.
   This panel enables you to define a view for all procedures with the (LIKE)
   pattern you define.

```
 DB2 Admin ----------- DB2X Create View on SYSIBM.SYSROUTINES ------------ 18:39
 Command ===>

  CREATE VIEW

  Owner    ===>         >
  Name     ===>                    >

  AS SELECT *
    FROM SYSIBM.SYSROUTINES
    WHERE SCHEMA LIKE '
 Pattern  ===>          >  '

  WITH CHECK OPTION ;

  GRANT SELECT,INSERT,UPDATE,DELETE ON (above table) TO
  Grantees ===>
```

Figure 434. Create View on SYSIBM.SYSROUTINES panel (ADB2ZF6)

2. Fill in the fields on this panel to create a view, for example, Define view
   ABC.FUNCTIONS as a view on SYSIBM.SYSROUTINES WHERE SCHEMA
   LIKE 'ABC. View ABC.FUNCTIONS contain all user-defined functions in
   schemas starting with ABC. In addition, you can issue GRANT SELECT,
   INSERT, UPDATE, or DELETE statements on the view to a list of authorization
   IDs (grantees).

Displaying views of functions

You can display views of functions.

About this task

To display the views that exist on SYSIBM.SYSROUTINES:

Procedure

Select option 7 on the Manage Functions panel. The Tables, Views, and Aliases panel is displayed, as shown in the following figure. This panel displays the views that exist on SYSIBM.SYSROUTINES.
The panel being displayed is the same panel you get if you use option 1.T and

```
  DB2 Admin ----------- DB2X Tables, Views, and Aliases ----------- ------

  Commands: GRANT       ALL
  Line commands:
   C - Columns  A - Auth  L - List  X - Indexes  S - Table space  D - Database
   V - Views  T - Tables  P - Plans  Y - Synonyms  SEL - Select prototyping
   ? - Show all line commands


  Sel   Name               Schema   T DB Name  TS Name    Cols      Rows Chks C


  ----- ------------------ -------- - -------- -------- ------ ----------- ---- -
        PROCEDURES         ISTJE    V DSNDB06  SYSOBJ       79          -1  0
        FUNCTIONS          ISTJE    V DSNDB06  SYSOBJ       79          -1  0
```

Figure 435. Tables, Views, and Aliases panel showing views on SYSIBM.SYSROUTINES (ADB21T)

option Z.PM.7.

Backing up and recovering a DB2 subsystem

The DB2 subsystem can be backed up, and jobs can be set up to specify a particular point in time to which to recover the subsystem or to recover the DB2 subsystem to a point in time.

Subsystem-level backups and recovery are possible only with DB2 for z/OS Version 8 or later, which supports the BACKUP SYSTEM and RESTORE SYSTEM utilities. Both utilities invoke z/OS DFSMShsm (Version 1 Release 5 or above). The BACKUP SYSTEM utility uses copy pools, which are new constructs in z/OS DFSMShsm. The RESTORE SYSTEM utility uses data that is copied by the BACKUP SYSTEM utility, and the data sets that are to be recovered must be SMS-managed data sets.

You can submit the batch job that DB2 Admin creates for backing up the system directly from DB2 Admin. You cannot directly submit the other batch jobs that DB2 Admin creates for specifying a particular time to which to recover the subsystem or for recovering the subsystem. These batch jobs cannot be run from DB2 Admin.

Topics:
- "Backing up the DB2 subsystem"
- "Specifying a point in time to which to recover" on page 545
- "Recovering the DB2 subsystem" on page 546

Backing up the DB2 subsystem

You can back up the DB2 subsystem.

About this task

To back up the DB2 subsystem:

Procedure

1. Select option SB on the System Administration panel. The Generate Backup panel is displayed, as shown in the following figure.

```
DB2 Admin----------------------- DB2X System Backup------------------- 20:24


  DSN of System Backup JCL . .
  Member name  . . . . . . .

  Backup Scope . . . . . . .          (F-Full, D-Data only)

  FORCE  . . . . . . . . . .          (Yes/No)
  DUMP   . . . . . . . . . .          (Yes/No)
    DUMPCLASS . . . . . . . .      >  (Up to 5 dump classes)
    FORCE . . . . . . . . . .         (Yes/No)
  DUMPONLY . . . . . . . . .          (Yes/No)
    TOKEN  . . . . . . . . .                              (Hex string)
    DUMPCLASS . . . . . . . .      >  (Up to 5 dump classes)

  BP - Change batch job parameters specified
```

Figure 436. System Backup panel (ADB2ZSB)

2. Enter the name of the data set and member in which the generated JCL is to be stored and specify copy options (or backup scope). Depending on the level of DB2 that you are using, some of the fields on this panel might be hidden. See the online help for a description of the fields that are displayed.

3. Press Enter. DB2 Admin displays the generated JCL for the backup job.

4. Submit the JCL to have the system backed up.

Specifying a point in time to which to recover

You can set up a batch job that will specify a particular time to which to recover the DB2 system.

About this task

To set up a batch job that will specify a particular time to which to recover the DB2 subsystem:

Procedure

1. Select option PT on the System Administration panel. The Generate Backup panel is displayed, as shown in the following figure.

```
DB2 Admin----------------------- DB2X System Point In Time Recovery---- 21:04
Command ===>


 DSN for DSNJU003 JCL. . . . :
 Member name . . . . . . . . :

 RBA/LRSN  . . . . . . . . . :

 BP - Change batch job parameters
```

Figure 437. System Point in Time Recovery panel (ADB2ZSB)

2. Enter the name of the data set and member in which the generated JCL is to be stored and specify an RBA value as the point in time for recovery of a non-data sharing member and an LSRN value as the point in time for recover of a data sharing member.
3. Press Enter. DB2 Admin displays the generated JCL for the job, as shown in the following figure.

```
//* STEP PITBKUP: RUN POINT-IN-TIME BACKUP
//*************************************************************
//PITBKUP EXEC PGM=DSNJU003
//STEPLIB  DD DISP=SHR,DSN=USER.TESTLIB
//         DD DISP=SHR,DSN=DSN810.SDSNLOAD
//SYSUT1   DD DISP=SHR,DSN=BSDS01
//SYSUT2   DD DISP=SHR,DSN=BSDS02
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
  CRESTART CREATE,SYSPITR=BBBBBBBB
/*
```

Figure 438. Example of Generated JCL for System Point in Time Recovery

4. Edit the generated JCL to specify the appropriate BSDS data set names in //SYSUT1 and //SYSUT2.
5. Save the JCL for the batch job. The batch job cannot be submitted directly after being created. It cannot be run from DB2 Admin.

Recovering the DB2 subsystem

You can set up a batch job that will recover the DB2 subsystem to a previous point in time.

About this task

To set up a batch job that will recover the DB2 subsystem to a previous point in time:

Procedure

1. Select option SR on the System Administration panel. The System Restore panel is displayed, as shown in the following figure.

```
DB2 Admin------------------------ DB2X System Restore------------------ 21:31
Command ===>


 DSN for Restore System JCL
 Member name  . . . . . . . .

 LOGONLY  . . . . . . . . . .              (Yes/No)
   SWITCH VCAT. . . . . . . .              (Yes/No)
     SYSVALUEDDN. . . . . . .              (DD name of VCAT alias data set)
 FROMDUMP . . . . . . . . . .              (Yes/No)
   DUMPCLASS  . . . . . . . .              (DFSMShsm dump class to use)
   RSA. . . . . . . . . . . .            > (DFSMShsm key-label to use)
   TAPEUNITS. . . . . . . . .              (Yes/No)
     Number of tape units . .              (Number of tape units to use)

 BP - Change batch job parameters specified
```

Figure 439. System Recovery panel (ADB2ZSR)

2. Enter the name of the data set and member in which the generated JCL is to be stored and specify appropriate options.

   Depending on the level of DB2 that you are using, some of the fields on this panel might be hidden. See the online help for a description of the fields that are displayed.

3. Press Enter. DB2 Admin displays the generated JCL for the job, which invokes the RESTORE SYSTEM utility.

4. Save the JCL for the batch job.

   Restriction: The batch job cannot be submitted directly after being created. It cannot be run from DB2 Admin.

Stopping DB2

You can stop the DB2 subsystem.

About this task

To stop the DB2 subsystem:

Procedure

1. Select option 2S on the System Administration panel. The Stop DB2 panel is displayed, as shown in the following figure.

```
DB2 Admin ----------------------- DB2X Stop DB2 ----------------------- 16:07
Command ===>

 -STOP DB2

  MODE(
  Stop mode        ===>            (Quiesce or Force, default is quiesce)
)
Note: After using FORCE mode, exit from DB2 Admin without issuing any further
SQL statements.
```

Figure 440. Stop DB2 panel (ADB2Z2S)

2. Enter Quiesce or Force in the **Stop mode** field.

3. Press Enter to stop DB2. DB2 Admin accomplishes this task by issuing the DB2 -STOP DB2 command.

The information that DB2 Admin returns to you from the command is in ISPF browse format.

Chapter 20. Managing space

DB2 Admin manages space by displaying DB2 and VSAM statistics for DB2 page sets and by invoking functions against objects.

Using the DB2 Admin Space Manager panels, you can:

- Display DB2 and VSAM information about DB2 page sets and invoke functions against objects. The statistical data is gathered from the DB2 catalog and merged with data from the VSAM catalogs.
- Alter page set properties
- Resize page sets to eliminate extents and free unused space
- Change to and from STOGROUP- and VCAT-defined space
- Estimate primary and secondary space allocation for new table spaces or indexes

Topics:

Restriction: The following limitations apply to the DB2 Admin Space Manager:

- The resize function generates separate jobs for each page set that exceeds the limits specified (primary command RESZ). This means that an index is reorganized twice, first by reorganizing the table space and then by reorganizing the index if the criteria for resizing are met by both spaces. Only the specific job for the index will update the allocations for the index.
- Resize calculations are based on the High Used RBA for the VSAM data set that contains the table space or index. This means that if activity on tables has left freespace in the pages, resize might overallocate space. This can be verified by repeating the resize. DB2 Admin Space Manager displays the message "No changes" if all selected spaces conform to the limitations given (number of extents, % used).

Launching DB2 Admin Space Manager

You can launch DB2 Admin Space Manager.

About this task

To launch DB2 Admin Space Manager:

Procedure

1. Select option SM on the Administration Menu panel. The Space Manager menu is displayed, as shown in the following figure.

```
 DB2 Admin --------------------- DB2 Space Manager --------------------- 16:33
 Option ===>

    1 - Display page set space by database            DB2 System: DB2X
    2 - Table space estimator                         DB2 SQL ID: ISTJE
    3 - Index space estimator

 For option 2 (optional):

 Table space name  . . . .                  (? to look up)
 In database . . . . . . .                  (? to look up. Default DSNDB04)

 For option 3 (optional):

 Index name  . . . . . . .          >  (? to look up)
 Schema  . . . . . . . . .          >  (Default ISTJE)


 Switch catalog copy . . . N (N/S/C)
```

Figure 441. The Space Manager menu (ADB2M)

2. Select one of the following options:

   1 — Display page set space by database
   Select this option to:

   - Display statistics for a page set.
   - Resize a page set to eliminate extents and to free unused space.
   - Switch between STOGROUP and VCAT-defined space.

   2 — Table space estimator
   Select this option to estimate the space that is required for a table.

   3 — Index space estimator
   Select this option to estimate the space that is required for an index.

   Switch catalog copy
   Select the catalog copy to use:

   N   No change. Continue using the same catalog.

   S   Switches to the local DB2 system catalog.

   C   Switches to a copy of the catalog or to a catalog at a distributed site. The Select Copy of DB2 Catalog panel is displayed, on which you can choose a copy of the catalog to use.

Displaying page set statistics

You can display page set statistics in various formats and issue a command against space-related objects.

About this task

To display page set statistics:

Procedure

1. Select option 1 on the Space Manager menu. The Space Management by Database panel is displayed, as shown in the following figure.

```
 DB2 Admin ------------ DB2X Display Pageset Space by Database ----------- 16:47
Option ===>

Enter the partial name of the database you want to display space statistics
for:

  Partial database name  . . : %          (required)
  Partial space name     . . : %          (optional)

  Partial owner name     . . : %       >  (optional)
  Partial VCAT name      . . : %          (optional)
  Partial storage group  . . : %       >  (optional)

  Include spaces . . . . . . : A          (All,indeXes, or tableSpaces)
```

Figure 442. Space Management by Database panel (ADB2M1)

2. Enter the following information:
   - Enter a partial database name. To improve performance, specify as much of the database name as possible.
   - Optionally, enter a partial space name. To improve performance, specify as much of the space name as possible.
   - Enter a partial owner name.
   - Enter a partial VCAT name.
   - Enter a partial storage group name.
   - Specify the type of spaces to be displayed.
     - Enter A to display both index and table space data.
     - Enter X to display index data.
     - Enter S to display table space data.
3. Press Enter. The Page Set Statistics for VSAM Statistics panel is displayed, as shown in the following figure. This panel contains VSAM-related page set data.

```
 ADB2M1S n ------------------- DB2X Page Set Statistics ----------- Row 8 of 18
Command ===>                                               Scroll ===> PAGE

Commands: VDEF VSTAT DDEF DSTAT RESZ
Line commands:
 I - Info  S - Space  SP - Space Part  G - Storage Group  DIS - Display
 STA - Start  STO - Stop  LISTC - Listcat  LISTD - Listcat Data
 AL - Alter  MOVE - Move  VDEF - VSAM define statement
 RESZ - Resize page set  HR - HSM recall  HL - HSM list  UT - Utility
       ? - Show all line commands

      Data      Page            Sub      VSAM          VSAM Pct VSAM
 Sel  Base      Set     Num T  Type KB Alloc   KB Used Usd Exts Volser #V
      *         *        * *   *        *             *   *    * *     > *
----- -------- -------- ---- -- ---- -------- ----------- --- ---- ------ --
      DSN8D61A DSN8S61D  1 S  SEG        48          48 100    1 RE9M01  1
      DSN8D61A DSN8S61E  1 S  LOB       144         144 100    3 RE9M05  1
      DSN8D61A DSN8S61E  2 X  IAUX      144         144 100    3 RE9M03  1
      DSN8D61A DSN8S61E  3 S            48           48 100    1 RE9M08  1
      DSN8D61A DSN8S61E  4 SP           144         144 100    3 RE9M05  1
      DSN8D61A DSN8S61P  1 S  SEG       192          96  50    1 RE9M10  1
      DSN8D61A DSN8S61R  1 S  LOB        48          48 100    1 RE9M10  1
      DSN8D61A DSN8S61S  1 S  LOB        48          48 100    1 RE9M05  1
      DSN8D61A XACT1     1 X  XML        48          48 100    1 RE9M08  1
```

Figure 443. Page Set Statistics panel (ADB2M1S) for VSAM statistics

You can focus on another area of page set statistics by issuing any of the following commands:
- VDEF to display VSAM definitions for the page data set

- DSTAT to display DB2 statistics for the page data set
- SDEF to display DB2 definitions for the page data set
- LISTC, LC, LIST, or LD to go to panel ADB2LCAT, TSO LISTCAT Output Display.

For more information about the fields that comprise these panels, see the online help. To display the VSAM Statistics panel again, issue the VSTAT command.

4. Use line commands to perform various space-related functions.

Resizing page sets

You can resize page sets in order to eliminate extents and to free unused space.

About this task

DB2 Admin Space Manager enables you to resize all page sets for a database or to select specific page sets to resize. The following instructions describe both methods.

Procedure

1. Complete steps 1 through 3 of "Displaying page set statistics" on page 550. The Page Set Statistics for VSAM statistics panel is displayed.
2. Resize all page sets or a specific page set.
   - If you want to resize all page sets, issue the RESZ primary command and press Enter.
   - If you want to resize a specific page set, tab to the page set that you want to resize and issue the RESZ line command and press Enter.

   The Resize Page Sets panel is displayed, as shown in the following figure. If the page set cannot be resized (because it is not overallocated or in extents), DB2 Admin issues a messages that indicates that there is nothing to resize.

```
DB2 Admin -------------------- DB2X Resize Page Sets -------------------- 20:50
Option ===>

Resize pagesets having:
  No. of extents greater than ===> 30  (1-100)
  Pct. used less than         ===> 90  (5-90)

  BP - Change batch job parameters
```

Figure 444. Resize Page Sets input panel (ADB2M1R)

3. Specify the following information:
   - In the **No. of extents greater than** field, specify the minimum number of extents that a page set must have in order to be resized, or
   - In the **Pct. used less than** field, specify the percentage of space that must be available for a page space to be resized. For example, if you enter 45 in this field, only those page sets that are using less than 45 percent of the space available are resized.
4. Press Enter. DB2 Admin creates a batch job to resize those page sets that meet the criteria that you specified.
5. Submit the job to resize the page sets.

Moving between STOGROUP- and VCAT-related space

You can move a page set that is currently in a STOGROUP-defined space to a VCAT-defined space on another volume

About this task

You can also move a page set that is currently in a VCAT-defined space to a STOGROUP-defined space.

If you enter the MOVE line command, you are prompted for additional input. The input asked for depends on whether you wish to move a STOGROUP-defined or a VCAT-defined page set.

To move between STOGROUP- and VCAT-related space:

Procedure

1. Complete steps 1 through 3 of "Displaying page set statistics" on page 550. The Page Set Statistics for VSAM statistics panel is displayed.
2. Tab to the page set that you want to move and issue the MOVE line command.
3. In the panel that displays, enter additional information. If you are moving a STOGROUP-defined page set, the Move Page Set Input panel (ADB2M1M) is displayed, as shown in the following figure.

```
DB2 Admin --------------- DB2 Space Manager Move Page Set --------------- 20:50
Option ===>

  1 - Move page set to another STOGROUP (with new VCAT)
  4 - Move page set from STOGROUP to VCAT

 New STOGROUP ===>         (current STOGROUP: DSN8G610 with VCAT: C1DB2)
 New Vcat     ===>         (for option 4)
 New volumes  ===>
```

Figure 445. Move Page Set input panel (ADB2M1M): STOGROUP-defined page sets

Option 1

    If you select Option 1, Move page set to another STOGROUP (with new VCAT), you must enter the names of the new storage group, and optionally that of a new catalog.

    New STOGROUP

        Specify the name of the new storage group. The name of the current storage group and VSAM catalog are displayed for your information.

    New VCAT

        Specify the name of a VSAM catalog.

Option 4

    If you select Option 4, Move page set from STOGROUP to VCAT, you must enter the name of a new VSAM catalog, and optionally, the new volumes for the page set. Use commas to separate volume names.

    New VCAT

        Specify the name of a VSAM catalog.

    New VOLUMES

        Optionally, specify the name of a new volume. For multiple volumes, separate the volume names with a comma.

If you are moving a VCAT-defined page set, the Move Page Set Input panel (ADB2M1M) is displayed, as shown in the following figure.

```
DB2 Admin --------------- DB2 Space Manager Move Page Set --------------- 20:50
Option ===>

   2 - Move page set to another VCAT
   3 - Move page set to other volume(s)
   5 - Move page set from VCAT to STOGROUP

 New STOGROUP ===>          (for option 5)
 New VCAT     ===>          (current VCAT: C1DB2)
 New volumes  ===>
```

Figure 446. Move Page Set input panel (ADB2M1M): VCAT-defined page set

Option 2
> If you select Option 2, Move page set to another VCAT, you must enter the name of the new VCAT, and optionally, the new volumes for the page set.
>
> **New VCAT**
> > Specify the name of a VSAM catalog. The name of the current VCAT is displayed for your information.
>
> **New VOLUMES**
> > Specify the name of a new volume. For multiple volumes, separate the volume names with a comma.

Option 3
> If you select Option 3, Move page set to other volume(s), enter the name(s) of one or more volumes.
>
> **New volumes**
> > Specify the name of a new volume. For multiple volumes, separate the volume names with a comma.

Option 5
> If you select Option 5, Move page set from VCAT to STOGROUP, enter the name of a new STOGROUP.
>
> **New STOGROUP**
> > Specify the name of the new storage group.

Table Space Estimator panel

You can use the DB2 Admin Space Manager to estimate the space requirements for a table.

About this task

To estimate the space requirements for a table:

Procedure

1. Select option 2, Table space estimator, on the Space Manager menu. The Table Space Estimator panel is displayed, as shown in the following figure. Initially, all of the fields on the Table Space Estimator panel are blank.

```
  ADB2MES n ----------------- DB2 Table Space Estimator ------------------ 18:33
 Option ===>

 Input values:
   No. of rows . . . . 100000       (required)
   Avg. row size . . . 100          (required, 1-32714)
   Page size . . . . . 4            (4,8,16, or 32, optional, default 4)
   Max rows/page   . . 255          (1-255, optional, default 255)
   Compression ratio . 0            (0-100, optional, default 0)
   Pctfree . . . . . . 5            (0-99, optional, default 5)
   Freepage. . . . . . 0            (0-255, optional, default 0)
   Segment size. . . . 0            (0 or 4,8,..,64, optional, default 0)
   Unit type . . . . . 3390         (3380/3390, default 3390)
   EAV support . . . . NO           (Yes/No, default No)

 Estimates:                         Suggested:
   Usable page size. : 3870           Primary  . . . . : 11520
   Rows per page . . : 35             Secondary  . . . : 1440
   Pages used  . . . : 2858         Disk estimates:
   Total pages . . . : 2860           Number of trks  . : 239
   Number of KB  . . : 11440          Number of cyls  . :16
```

Figure 447. Table Space Estimator panel example (ADB2MES)

2. Fill in the fields in the Input values section of the panel.
3. Press Enter. The Table Space Estimator panel is displayed again. Based on the input values you entered, the Table Space Estimator provides information about the estimated space that the table will require and suggests the amount of space that you should allocate for this table.

   For the **Compression** field, the value represents the percentage of rows that will not be compressed. For example, a compression value of 1 yields the maximum compression (because 99% of the rows are compressed). A compression value of 99 yields the minimum compression (because only 1% of the rows is compressed). A value of zero represents zero compression.

Index Space Estimator panel

You can use the DB2 Admin Index Space Estimator to estimate the index space requirements for a table.

About this task

To estimate the index space requirements for a table:

Procedure

1. Select option 3, Index space estimator, on the Space Manager panel. The Index Space Estimator panel is displayed, as shown in the following figure. Initially, all of the fields on the Index Space Estimator panel are blank.

```
ADB2MEX n  --------------    DB2 Index Space Estimator    ------------- 18:46
Command ===>

Input values:
  No. of keys  . . . .              (required)
  Key length  . . . .               (required, 1-2000)
  Unique  . . . . . .               (required, Yes/No)
    Distinct  . . . .               (for non-unique: no. of distinct keys)
    OR rows/key . . .               (for non-unique: avg. rows per key)
  Compression ratio . 0             (0 or 12.5-100, optional, default 0)
  Page size . . . . . 4             (4, 8, 16, or 32, default 4)
  Pctfree . . . . . .               (0-99, default 5)
  Freepage  . . . . .               (0-255, default 0)
  Large TSpace  . . .               (Yes/No, default No)
  Unit type . . . . . 3390          (3380/3390, default 3390)
  EAV support . . . . NO            (Yes/No, default No)
  No. of pieces . . .               (1-32, 1-4096 with large table space)
  OR piecesize  . . .               (nX, n=numeric value, see help,X=K/M/G)

Estimates:                       Suggested:
  Usable page size  :                Primary  . . . . :
  Keys per page . . :                Secondary  . . . :
  Leaf pages  . . . :                Piecesize  . . . :
  Index levels  . . :              Disk estimates:
  Total pages . . . :                Number of trks . :
  Number of KB  . . :                Number of cyls . :
```

Figure 448. Index Space Estimator panel (ADB2MEX)

2. Fill in the fields in the Input values section of the panel.

3. Press Enter. The Index Space Estimator panel is redisplayed. Based on the input values you entered, the Index Space Estimator provides additional information about the estimated space that the table will required and suggests the amount of space that you should allocate for this table.

   The following fields are available on the panel. The first three fields are required.

   No. of keys
   The number of keys in the index that refer to data rows.

   Key length
   The sum of the length of all the columns of the key, plus the number of the columns that allow nulls.

   Unique
   Specify whether the key is unique. 'NO' means non-unique.

   Distinct
   For a non-unique index: number of distinct keys. If specified it will be used to calculate the average number of rows per key. Can not be specified if "Rows/key" is specified.

   OR rows/key
   For a non-unique index: average number of rows per distinct key. Cannot be specified if "Distinct" is specified.

   The remainder of the fields are optional.

   Page size
   Specifies size of the pages in KB. The default is 4 KB.

   Pctfree
   The percentage of each page to leave as free space when the table is loaded or reorganized. The default is 5 percent.

Freepage
Specifies how often DB2 will leave a page of free space when the table is loaded or reorganized.

Large TS
Specifies whether the table space used by this index is defined as LARGE.

Unit type
Unit type to be used when calculating the estimated number of tracks and cylinders.

EAV
If Extended Address Volume (EAV) parameter is set to YES, the space estimate is increased by 10 cylinders and then rounded up to a multiple 21 cylinders.

No. of pieces
Number of data set pieces into which to split the index. When you specify a value and press Enter, the Suggested Piecesize field is calculated and displayed.

OR piecesize
Value in kilobytes (K), megabytes (M), or gigabytes (G). The suggested number of pieces is calculated and displayed. Example values include: 1024M, 1G, and 4096K. Valid values for n are:

K 256, 512, 1024, 2048, 4096, 8192, 16384, 32768, 65536, 131072, 262144, 524288, 1048576, 2097152, and 67108864.

M 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192, 16384, 32768, and 65536.

G 1, 2, 4, 8, 16, 32, and 64.

The remainder of the panel consists of estimates and recommendations generated by the index space estimator.

The output fields are:

Usable page size
The number of bytes per page that can be used for rows.

Keys per page
The number of keys per leaf page.

Leaf pages
The number of leaf pages.

Index levels
The number of index levels.

Total pages
The total number of pages in the index. Includes header pages, space map pages, and free pages.

Number of KB
The estimated number of KB required for this index.

Primary quantity
The suggested primary quantity for this index in KB.

Secondary qty
The suggested secondary quantity for this index.

Piecesize
The suggested piece size when number of pieces is specified. The default value for Piecesize is 2G (2 gigabytes).

Number of trks
The estimated number of tracks required.

Number of cyls
The estimated number of cylinders required.

Example

The following figure shows a second view of the Index Space Estimator panel. Assuming that the values have been entered in the fields, the space estimator generates the estimates shown in the lower portion of the panel. Both input and output values are displayed on the panel.

```
ADB2MEX n  --------------    DB2 Index Space Estimator   ------------- 18:46
Command ===>

Input values:
  No. of keys . . . . 100000      (required)
  Key length  . . . . 10          (required, 1-2000)
  Unique  . . . . . . Y           (required, Yes/No)
    Distinct  . . . .             (for non-unique: no. of distinct keys)
    OR rows/key . . .             (for non-unique: avg. rows per key)
  Compression ratio . 0           (0 or 12.5-100, optional, default 0)
  Page size . . . . . 4           (4, 8, 16, or 32, default 4)
  Pctfree . . . . . . 5           (0-99, default 5)
  Freepage  . . . . .             (0-255, default 0)
  Large TSpace  . . . NO          (Yes/No, default No)
  Unit type . . . . . 3390        (3380/3390, default 3390)
  EAV support . . . . NO          (Yes/No, default No)
  No. of pieces . . .             (1-32, 1-4096 with large table space)
  OR piecesize  . . . 256K        (nX, n=numeric value, see help,X=K/M/G)

Estimates:                    Suggested:
  Usable page size  : 3836      Primary  . . . . : 1824
  Keys per page . . : 225       Secondary  . . . : 48
  Leaf pages  . . . : 445       Piecesize  . . . : 256 K
  Index levels  . . : 3       Disk estimates:
  Total pages . . . : 450        Number of trks . : 38
  Number of KB  . . : 1808       Number of cyls . : 3
```

Figure 449. Index Space Estimator panel example (ADB2MEX)

Chapter 21. Managing changes to DB2 objects

DB2 Admin manages and tracks the changes that you make to your DB2 objects.

Topics:

Overview of Change Management

The Change Management function in DB2 Admin simplifies the process of recording and tracking the changes that you make to your DB2 objects, which can be very complex, especially when others have defined changes that have yet to be run.

Change Management provides the following features:
- Assigns a change ID for every change that you make to your DB2 objects, and registers each change in the Change Management database
- Enables you to analyze how a change affects existing objects
- Warns you if there are pending changes to the same object that you plan to change, which gives you the opportunity to specify whether your change should supersede or follow the pending changes
- When pending changes exist for the objects, allows you to define the new changes as if the pending changes have already been made
- Facilitates the generation of new versions to provide a snapshot of your database definitions after changes have been made
- Uses an interface that allows you to track and query changes to objects and quickly find all of the components that are involved in a change
- Provides an audit trail and helps automate the process of recovering changes
- Maintains the relationships between changes, versions, masks, ignores, generated DDL, and unloaded data
- Facilitates moving changes from one DB2 subsystem to another

You can use Change Management for changes that you make by using the following DB2 Admin or DB2 Object Comparison Tool features:
- SQL CREATE, ALTER, DROP, RENAME, COMMENT, and LABEL statements that are executed from the input screen or from a data set and SQL REVOKE statements that are executed from the input screen or from a data set as immediate changes

- The AL line command to change or rename a database
- The AL line command or ALT command to change a table space or index space
- The AL line command or ALT line command to change a table
- Comparisons in which changes are made to synchronize the target system with the source system
- Changes that are defined through the Change Management panels

You need the DB2 system parameter (DSNZPARM) values to write the version file. Specify the input option GETDB2ZP='Y' in the Change DB2 Admin Defaults panel so that GEN calls the DB2 stored procedure DSNWZP to get the DB2 system parameter (DSNZPARMS) values.

Restriction: The GRANT USAGE ON JAR statement is not supported in the DB2 Object Comparison Tool.

Change Management terminology

Understanding the terminology that is involved in managing changes will help you use Change Management.

The following terminology will help you use Change Management:
- *Exclude specification* is a list of objects that you specify to be omitted from the compare process. The selected objects are not included as input or output of the compare process.
- A *fast* change is a change that can or should be run immediately. If the affected objects have pending changes, the fast change is called an *emergency change*, and it will supersede the pending changes. If the affected objects do not have any pending changes, the fast change is called an *immediate change*.
- *Ignore changes specification* is a list of changes to objects from saved compare results that you specify to be ignored in subsequent compare processing. The selected object types participate in the compare process but changes to the object types are not propagated.
- *Ignore fields* specify the DB2 catalog fields that should be ignored when objects are compared.
- *Masks* (or *translation masks*) specify how names are to be translated when objects are compared or when they are moved from one system to another (source to target). Masks also allow you to overwrite the values of certain table space and index space attributes.
- A *multi-target change* is a change that is initially registered on one system (the "central" system), and that can be used to distribute and track a change to database objects across one or more target systems. A separate change is registered and runs on each target system.
- A *prerequisite change* is a change that must be run before the current change is run. When you create a change for an object, the object might have *pending changes*, which are changes that have yet to be run. You can choose to make the pending changes remain as prerequisite changes for the new change or make your new change a *supersede change*, which puts the new change ahead of the pending changes.
- A *recover change* is a change that lets you back out a change that has been completed.
- Backing out a completed change requires determining whether the change has a recover change, whether other changes must be recovered first and in which order, and whether there are pending changes to the objects being affected by

the change that will have to be reanalyzed after the change is recovered. DB2 Admin uses a *recover strategy* to determine all of this information for you.

- A *version* is a snapshot of a set of object definitions at a point in time. With Change Management, you have the option of creating a new base version before or after applying a change. You can then use this *base version* for a subsequent change or choose to generate DDL from the base version.

  In Change Management, the special type of version file that is called a *delta version*, is no longer used.

- A *version scope* identifies the set of objects to be included in processing a version. It determines the objects that will be included in a version.

The Change Management process

The most common tasks that you need to perform when you use Change Management to make changes are defining the changes, registering the changes, analyzing the changes, and running the changes.

Defining a change

Changes can come from a variety of sources. For example, you can use the regular features of DB2 Admin or DB2 Object Comparison Tool to generate DB2 object changes, import statements into a change from a data set, or import changes that have been promoted from other DB2 subsystems.

The objects that you are changing might have pending changes, which are changes that are being made through Change Management and have yet to be completed. You will need to specify whether the change that you are defining should supersede these pending changes or not. When and where you specify how pending changes should be handled depends on whether the pending changes can be applied as virtual changes:

- When the pending changes can be applied as virtual changes, you specify how to handle the pending changes at the time you define the change.

  When you define the change, the list of pending changes is displayed, and you must specify whether to apply these pending changes and define your new change based on a virtual representation of the objects with the pending changes applied. When you choose to supersede the pending changes, you define your change without taking the effect of the pending changes into account, and the new change becomes a prerequisite change for the pending changes.

  Pending changes can be treated as virtual changes when you make changes by using the following methods:
  - The Alter dialogs to:
    - Rename a database (ALT)
    - Redefine a table space (ALT)
    - Redefine an index (ALT)
    - Redefine a table (ALT)
  - The Create dialogs (option 2.4 from the DB2 Admin main menu) to create a table space, table, index, materialized table, view, and trigger
  - The Tables, Views, and Aliases system catalog panel (option 1.T from the DB2 Admin main menu) to rename a table

  For performance reasons to minimize the amount of time spent traversing relationships, especially for renames, the list of pending changes that DB2 Admin displays might not be complete. However, if you apply the pending changes, all of pending changes for the objects are applied whether or not they appear in the list.

Tip: To minimize the amount of time that it takes to apply pending changes, keep the number of uncompleted changes (DEFINED, ANALYZED, RUNNING) to a minimum.

- When the pending changes cannot be applied as virtual changes, you are prompted to specify how the pending changes should be handled at the time you register the change (or shortly before you register the change when the source of the change is from importing statements into a change from a data set, importing changes that have been promoted from other systems, or performing a comparison in DB2 Object Comparison Tool). You have to determine whether your new change should be added to a pending change, be made before or after any pending changes that exist, or be executed immediately.

  Exception: You are not prompted to specify how to handle pending changes if you use the Change Management panels to define a change (that is, insert a change on the Changes panel and then create change statements for the change)

Registering a change

After you define a change, DB2 Admin prompts you to register the change in the Change Management database. After specifying a name for the change, DB2 Admin automatically assigns a change ID to the change.

Depending on the method that was used to define the change, you might be prompted to specify whether to register the change as a normal change, a multi-target change, an emergency change, or an immediate change and how to handle pending changes for the objects that are involved in the change:

- If there are pending changes, you can register the change as a normal change or an emergency change. If you register the change as a normal change, you also must specify whether the change should be made before or after the pending changes.
- If there are no pending changes, you can register the change as a normal change or an immediate change.

DB2 Admin runs emergency and immediate changes immediately. The analyze and run phases do not apply.

Analyzing a change

A normal change must be analyzed before the change can be applied to the objects. When you issue the command to analyze a change, DB2 Admin generates a batch job that you submit.

The batch job analyzes how the change modifies existing objects, both in DB2 and in any of the prerequisite changes, and creates a work statement list (WSL) that will be used to run the changes. During the analyze process, the embedded SQL statements semantics are checked and DB2 Admin automatically generates two new base versions:

- A target version, which represents the objects in the DB2 catalog plus any prerequisite changes
- A source version, which is the target version plus the changes for the change that being analyzed

DB2 Admin then invokes DB2 Object Comparison Tool to compare the source and target base versions to generate a WSL that will be used in the run process to apply the changes. The base versions that are used in this process are temporary and are not saved.

DB2 Admin generates the base version using one of the following methods:

Automatic (A)

The base version is generated from the DB2 catalog using the objects that are referenced in the change.

User-defined (U)

The base version is generated from the DB2 catalog using the objects that are specified in the version scope.

Existing (E)

An existing base version is used. DB2 Admin uses the current contents of the existing version and the contents of the DB2 catalog are not considered.

You can specify the method that DB2 Admin uses when there are no prerequisite changes for the change. If prerequisite changes exist, DB2 Admin chooses the method based on the following criteria:

User-defined (U)

This method is forced if all the prerequisites have a status of ANALYZED and use the same version scope. The same version scope will be used for the change you are analyzing.

Existing (E)

This method is forced if all the prerequisites have a status of ANALYZED and use the same base version. The same base version will be used for the change you are analyzing.

Automatic (A)

This method is forced if neither of the previous conditions are true.

When you analyze a change, you can specify that a recover change be created automatically. Creating a recover change gives you the option of backing out the change. When you first choose to create a recover change, you are prompted to register the recover change. The recover change is automatically updated if the original change is reanalyzed.

Running a change

After a change has been successfully analyzed, it is ready to be run. That is, you are ready to apply the change to the database. When you issue the command to run the change, DB2 Admin creates a batch job that you submit. The batch job runs the WSL that was generated during the analyze process.

If the change has prerequisite changes, you cannot run the change, and DB2 Admin will prompt you to run the prerequisite changes first.

The run job performs a task called *runtime analyze*, which ensures that the DB2 catalog has not changed from the time the change was analyzed. The run job reanalyzes the change and creates a second WSL, using the current DB2 catalog and the automatic base version method. The second WSL is compared with the WSL that was generated during the normal analyze. If the DDL and DCL content are the same, the run job continues and the WSL that was generated during the normal analyze is run to apply the changes. If the DDL or DCL content are different, the run job stops with a return code of 8, and the change is not run.

When you run a change, you have the option of specifying that a new base version be created after the changes have been made successfully. If you want to have a new base version created, a version scope that defines the set of objects to be included in the base version must exist.

Requirement: Always use the RN command on the CM - Changes panel (ADB2C11) to run changes that are being managed under Change Management. Do not run the WSL that was generated during the analyze process directly from the Work Statement List Library panel (ADB2W1) because DB2 Admin cannot track changes that are made outside of the Change Management process. Also, do not use the line commands on the Work Statement List Library panel to edit, delete, copy, append, or clone a WSL that was generated during the analyze process.

You can also use Change Management to complete many other tasks. You can recover changes; track changes and changed objects; manage masks, ignores, versions, and version scopes; and promote changes from one system to another.

Types of changes and change status

To facilitate change management, DB2 Admin categorizes changes into several types and assigns a status to each change as it moves through the change management process.

The type is assigned when a change is registered. The following table describes the types of changes:

Table 17. Types of changes

Type of change	Description
CHANGE	A change that is defined through the usual change functions in DB2 Admin and DB2 Object Comparison Tool, such as ALT, and compare, which go through the change management process of being analyzed and then run.
MULTI-TC	A change that is generated on the central system for the purpose of importing on to multiple target systems. On target systems, the changes that are registered are then analyzed and run in order to apply the changes to the target catalog.
FAST	A change that is run immediately. If pending changes exist for the object or related objects that are affected by the fast change, the fast change is called an emergency change, and it supersedes the pending changes. The pending changes are placed in DEFINED status. If no pending changes exist, the fast change is called an immediate change. Because fast changes are run immediately upon registration, you cannot analyze or run them manually. You also cannot modify fast changes, recover them, or promote them to other systems.
COMPARE	A change that is generated by comparing two items such as two base versions, two DDL files, two catalog objects, or a DDL file and a catalog object.
PROMOTE	A change that is generated by importing statements from a data set or a changes file.
RECOVER	A change that was automatically generated to back out another change. When you analyze a change, you have the option of having a recover change created. DB2 Admin generates a recover change, assigns a change ID to the recover change, and puts the recover change in ANALYZED status. To recover a change, you issue the RC line command for the original change. You do not issue the RN line command for the recover change.

The status of a change is updated when actions are taken on the change. The following table describes the possible values for the status:

Table 18. Status of changes

Status	Description
INITIAL	The change has been created, but its registration in the Change Management database is incomplete. You can try to get the change registered by issuing the restart line command (RST) on the Changes panel (ADB2C11).
	If a change is in INITIAL status and you issue the restart line command to attempt to complete it, DB2 Admin cannot detect and process any prerequisite changes that might exist. You will need to identify any prerequisite changes yourself and reanalyze any change in ANALYZED status to ensure its validity.
DEFINED	The change has been created and registered in the Change Management database. The change is ready to be analyzed.
ANALYZED	The change has been validated and a WSL to run the change has been generated. The change is ready to be run.
RUNNING	The change is currently being run. A RUNNING status that does not change to COMPLETE status indicates that the job to run the change failed at some point.
COMPLETE	The change has been run successfully.
CANCELED	The change has been canceled.
FAILED	The change is a fast change that was run immediately but did not complete successfully.

The Change Management main menu panel

The Change Management (CM) panel, which can be accessed by using the CM option on the DB2 Administration Menu panel, is the main menu for accessing Change Management functions.

The Change Management (CM) panel is shown in the following figure:

```
DB2 Admin ------------------- Change Management (CM) ------------------ 19:27
Option ===>


   1 - Manage changes                              DB2 System: DB2X
   2 - Manage masks                                DB2 SQL ID: ISTJE
   3 - Manage ignores                              CM Owner  :  ADB
   4 - Manage versions
   5 - Manage ID table
   6 - Report changes
   7 - Manage exclude specifications
   8 - Manage ignore changes specifications
   9 - Manage targets
```

Figure 450. Change Management (CM) panel (ADB2C)

The following options are available on this panel:

Manage changes
Select this option to manage changes. From the Manage Changes panel, you can display changes to perform various actions such as analyzing the change, running the change, or recovering the change. You can also use this panel to

create a change, create a delta for a target location (promoting a change), or import a delta that was created (importing a change).

Manage masks
Select this option to manage masks. From the Manage Masks panel, you can display the masks that are defined or you can create a new mask.

Manage ignores
Select this option to manage ignores. From the Manage Ignores panel, you can display the ignores that are defined or you can create a new ignore.

Manage versions
Select this option to manage versions. From the Manage Versions panel, you can display versions and version scopes. You can also create a version scope.

Manage ID table
Select this option to change the default Change Management level or to override the default level for specific SQL IDs.

Report changes
Select this option to display changes or changed objects.

Manage exclude specifications
Select this option to create, edit or display exclude specifications.

Manage ignore changes specifications
Select this option to manage ignore changes.

Manage targets
Select this option to display or create targets for change.

Restriction: The value of the character input fields on the Change Management panels cannot contain an apostrophe (or single quotation mark). For example, do not specify an apostrophe in the name of any change, version, mask, or ignore.

Tip: You can issue the CMM special command from any DB2 Admin panel to go directly to the Change Management (CM) panel.

Prerequisites for Change Management

DB2 Object Comparison Tool Version 11 Release 1 must be installed to use Change Management, and DB2 Admin must have been customized so that Change Management is enabled.

The Change Management database manages several objects that are required by the product.

In addition, for you to be able to register changes in the Change Management database, either the default Change Management level or the level for the current SQL ID must be either REQUIRED or OPTIONAL. The Change Management levels are:

REQUIRED
All changes must be registered in the Change Management database.

OPTIONAL
Changes can be registered in the Change Management database. When you define a change, you are prompted as to whether to make the change through Change Management.

Changes to a set of objects that are being managed under Change Management should all be made through Change Management.

OPTIONAL might be used when you are testing Change Management or when you can ensure that the SQL ID will register the changes to objects that are being managed under Change Management when prompted.

NONE

No changes can be registered in the Change Management database.

Setting the default Change Management level

The default Change Management level in the Change Management ID table determines whether changes must be registered, can be registered, or cannot be registered in the Change Management database if a level has not been specifically defined for the current SQL ID.

About this task

To change the default Change Management level:

Procedure

1. On the Change Management (CM) panel, specify option 5 to display the Manage ID Table panel.
2. Change the default change management level setting to the desired value: REQUIRED, OPTIONAL, or NONE.
3. Issue the SAVE command to update the Manage ID Table.

Setting the Change Management level for specific SQL IDs

You can override the default Change Management level for one or more specific SQL IDs by defining an entry for the SQL IDs in the Change Management ID table.

About this task

To specify the Change Management level for a specific SQL ID:

Procedure

1. On the Change Management (CM) panel, specify option 5 to display the Manage ID Table panel.
2. Add a new SQL ID or change the Change Management level for an existing ID.
   - To add a new SQL ID, issue the I line command, and specify the SQL ID and the change management level for the SQL ID (REQUIRED, OPTIONAL, or NONE).
   - To change the Change Management level for an existing SQL ID, type over the current value in the Level column.
3. Issue the SAVE command to update the Manage ID Table.

Recommendations for designing a Change Management strategy

An effective change management strategy is one that is well planned. The most important factor to consider is to ensure that changes to a set of objects are either all performed through Change Management or are all performed without Change Management.

Requiring that all changes go through Change Management is easy when the objects that should go through Change Management are handled by a few SQL IDs and the SQL IDs are used only for these objects. If the SQL IDs are also being used to change objects that should not go through Change Management, you should set

the Change Management level option to OPTIONAL, and the user will have to decide whether the change should go though Change Management.

A few example Change Management strategies are:
- When Change Management is being used for the objects for only one application:
  - Set the Change Management level for the SQL ID that is used to manage the objects for the application to REQUIRED.
  - Set the level for the other SQL IDs to NONE by setting the default Change Management level to NONE.
- When Change Management is being used for the objects for all applications except for a few objects that are under design and development:
  - Set the default Change Management level to REQUIRED.
  - Set the Change Management level for the SQL IDs that are used to change the objects that are under design and development to NONE. If those SQL IDs are also used to change objects that are not under design and development, set the Change Management level for the SQL IDs to OPTIONAL; the user will need to specify whether to use Change Management upon each change.
- When Change Management is being tested:
  - Set the default Change Management level to OPTIONAL.

Change Management scenarios

Change Management scenarios illustrate how you might use Change Management to make a simple change to a database and move changes that are made on one system to another.

Topics:
- "Scenario: Making a simple change to a database"
- "Scenario: Promoting changes from one system to another" on page 569

Scenario: Making a simple change to a database

This scenario explains how to make changes to part of a database structure on a development system.

About this task

Specifically, for the EMP table, you want to drop the COMMISSION column and increase the length of the LASTNME column to 45 bytes.

In making these changes, you have the following goals:
- Ensure that there is a snapshot of the database structure for fallback purposes.
- For the dropped column, repair any side effects of the change, such as handling inoperative or undefined objects such as packages, views, and triggers.
- Ensure that data is preserved for the change to the column length.
- Optimize the database with respect to the changes, such as running RUNSTATS or rebinding where necessary.
- Capture the changes for auditing purposes.

The following steps show you how you might use Change Management to make these changes and achieve your goals:

Procedure

1. Generate operations to track the change in Change Management. Create a version scope of the human resources database. You want to define a version scope because you want to create a snapshot (or base version) of the database structure after the changes are made. The version scope defines the objects that should be in the base version.

2. Modify the length of the LASTNME column and drop the COMMISSION column.

   a. Find and select the EMP table.

   b. Issue the ALT command to change the table. If there are any pending changes to the table, specify whether to implement your changes based on the assumption that the pending changes have been performed or that they have not been made and your change should supersede them. In this scenario, assume that there are no pending changes.

   c. Type over the length of the LASTNME column to increase the length to 45.

   d. Issue the D line command to delete the COMMISSION column.

3. Identify the impact that the changes have. To assess the impact of increasing the length of the LASTNME column and dropping the COMMISSION column:

   a. Type the REL primary command to see the related objects.

   b. Select each related object individually and determine if any changes are required because of the change in length to LASTNME or for COMMISSION being dropped. In this example, assume that a view is impacted by the dropped column.

4. Repair the side effects for the change. To fix the view:

   a. Issue the A line command to change the view.

   b. In the edit session that is displayed, remove the predicate from the view and save the edit session. The new definition of the view will be included as part of the change.

   c. Issue the CONTINUE command to finalize the changes to the table and the view.

5. Register the change in the Change Management database. Change registration occurs in this scenario because Change Management is enabled and required. To register the change, specify an owner and name for the change.

6. Analyze the change. To analyze the change:

   a. Go to the Change Management main menu and display the list of changes.

   b. Issue the analyze command for the change.

   c. Submit the batch job that DB2 generates to perform the analyze. The batch job produces a report of the changes that will be made and generates a WSL that will make the changes.

7. Run the change and capture a snapshot of the database structure after the change is complete. To run the change:

   a. Go to the Change Management main menu and display the list of changes.

   b. Issue the run command for the change, specifying that a new base version of the database structure should be created after the changes are made.

   c. Submit the batch job that runs the WSL that applies the changes.

Scenario: Promoting changes from one system to another

This scenario supposes that you are asked to promote the changes that were made to the human resources database on the development system to the test system.

About this task

This task requires you to determine the differences between the development and test system and apply the changes to the test system.

Assume that versions for the current state of the databases exist (Release 11C in DEV and Release 11B in TEST). You will compare the two versions to generate a delta changes data set that contains the SQL statements that represent the differences, transfer the delta changes data set to the test system, import the delta changes data set on the test system as a new change, and then apply the changes to the test database.

In synchronizing the human resources database, you have the following goals:
- Ensure that there is a snapshot of the database structures for fallback purposes.
- Capture the changes that are made on the test system for auditing purposes.

The following steps show you how you might use Change Management to make these changes and achieve your goals:

Procedure

1. On the source system (the development system), use the current versions of the development and test databases to identify the differences between the databases and promote the differences to the target system (the test system). To find and promote the differences in a delta changes data set:
   a. Go to the Change Management main menu and display the Manage Changes panel.
   b. Select the option to create a delta changes file for the target system.
   c. Identify the version of the test database as the starting version and the version of the development database as the ending version. Provide a name for the job that will generate the delta changes data set and a name for the delta changes data set. The starting version is a snapshot of the objects before changes are made, and the ending version is a snapshot of the objects after changes are made. In this scenario, you want to bring the level of the test system up to the level of the development system.
   d. Register the change in the Change Management database. You will be prompted to register the changes that are being promoted.
   e. Submit the batch job that creates the delta changes data set.
2. Import the delta changes data set as a new change on the test system. To import the promoted changes on the test system:
   a. Go the Change Management main menu on the test system and display the Manage Changes panel.
   b. Select the option to import changes.
   c. Specify the name of the delta changes data set to import into a change, and register the imported change in the Change Management database. Importing a change is a two-step process. First, DB2 Admin performs an analysis to determine if there are any prerequisite changes that are pending for the objects that are affected by the imported change. Next, the change is registered. The steps can be performed either in the foreground (TSO) or the background (batch).
3. Analyze the imported change. To analyze the change:
   a. Go to the Change Management main menu and display the list of changes.
   b. Issue the analyze command for the change.

c. Submit the batch job that DB2 generates to perform the analyze. The batch job produces a report of the changes that will be made and generates a WSL that will make the changes.

4. Run the imported change and capture a snapshot of the test human resources database after the change is complete. To run the change:

   a. Go to the Change Management main menu and display the list of changes.

   b. Issue the run command for the change, specifying that a new base version of the database structure should be created after the changes are made.

   c. Submit the batch job that runs the WSL that applies the changes.

Making changes through Change Management

Making a change through Change Management consists of three steps.

Topics:
- "Registering a change"
- "Analyzing a change" on page 576
- "Running a change" on page 579

Registering a change

When you create a change and Change Management is required (or Change Management is optional and you have specified to use Change Management), DB2 Admin prompts you to register the change in the Change Management database.

About this task

To create and register a change:

Procedure

1. Define the change. For example, change a table by using the ALT command or run SQL statements from a data set or screen input.

   If DB2 Admin displays a list of pending changes for the affected objects in the Pending Changes - Conflict Resolution panel, specify whether to apply the pending changes as virtual changes before you continue to define your change.

   Tip: If Change Management is optional for your SQL ID, specify YES when you are prompted whether to use Change Management.

2. Fill in the fields on the Register Options panel, and issue the CONTINUE command.

3. Specify the following information:

   - Specify an owner and a name for the change. The default owner is the current SQL ID. If you specify the name of a existing change, the change statements are included in the existing change, if possible.

     You can include the change statements in an existing change when the existing change has no prerequisite changes and the existing change is not a recover change, a fast change, or a promote change on the source side.

   - Optionally, specify a comment for the change.

   - Specify if you want to replace an existing change. If you leave the field blank any existing change is not replaced.

   The following figure shows an example of the Register Options panel:

```
DB2 Admin ------------------- CM - Register Options ------------------- 21:36
Option ===>

Commands: CONTINUE                                    DB2 System: DB2X
                                                      DB2 SQL ID: JOHNSON
Specify the following values to register a change:

Owner . . . . . . . . JOHNSON  >  (Optional, default is JOHNSON)
Name  . . . . . . . . EMP_CH4                                      >
Comment . . . . . . . Increase the length of WORKDEPT             >

Replace existing change . .              ('/' to replace, Default is BLANK)

Specify the owner and name values to use for this change (? to lookup):
                     Owner        Name
Ignore  . . . . . . .          >                                  >
Mask  . . . . . . . .          >                                  >
```

Figure 451. Register Options panel (ADB2CRO)

> **Exception:** Depending on how the change was defined, you might first be prompted about how to register the change on the Register Change panel:
> - If there are no pending changes, you can register the change as an immediate change or as a normal change.
> - If there are pending changes, you can register the change as an emergency change or as a normal change. When you register it as a normal change, you must specify whether the pending changes should be prerequisite changes for the change or whether the change should supersede the pending changes.
>
> If you register the change as an immediate or emergency change, you specify an owner and name for the change, and optionally, a comment. When you press Enter, the change runs immediately. If you register the change as a normal change and press Enter, the panel in the previous figure is displayed.
>
> **Note:** For DB2 V9 or later versions, Register might insert SET CURRENT SCHEMA statements. If the first statement of the change is not a SET SCHEMA statement and if the value of CURRENT SCHEMA is different from the value of CURRENT SQLID, register will insert a SET SCHEMA statement into ADBCHGS prior to processing other given statements. And the LASTSCHEMA column of change table ADBCHG will be updated with the current SCHEMA.
>
> **Note:** Also, when more statements are added to an existing change, the LASTSCHEMA will be checked against the current schema and, if they are different, another SET SCHEMA statement will be inserted by Register.
>
> **Restriction:** When Register is triggered via **Restart** or **Editing the change statements via CM panels**, SET SCHEMA statements will not be inserted by Register. However, the new column LASTSCHEMA in table ADBCHG will be updated.

4. Optional: Verify that the change was registered and is in DEFINED status by completing the following steps:
   a. Enter the CMM command to display the Change Management (CM) panel.
   b. Select option 1 to display the Manage Changes panel.
   c. Select option 1 to display the Changes panel.
   d. Verify that your change is included in the list of changes.
5. If the change is not registered successfully and is placed in INITIAL status, you can issue the restart line command (RST) to attempt to complete registration. However, when you restart the change, DB2 Admin cannot detect and process

any pending changes that might exist. You will need to identify any pending changes yourself and reanalyze any change in ANALYZED status to ensure its validity.

Results

DB2 Admin has registered your change in the Change Management database and has automatically assigned a change ID to it.

Example 1: Registering a change that is defined with the ALT command

This example shows how to register a change when pending changes can be applied as virtual changes before you define your change, such as when you use the ALT command to redefine a table.

Procedure

1. Issue the ALT command for the table that you want to change.
2. If Change Management is optional for your SQL ID, specify YES when you are prompted whether to use Change Management in the Change Management Prompt pop-up panel that is displayed.
3. If DB2 Admin displays a list of pending changes that exist for the affected object (that are registered in Change Management), specify how the pending changes are to be handled before the object definition is shown, and issue the NEXT command. The following figure shows an example of an object that has pending changes:

```
DB2 Admin ------------ Pending Changes - Conflict Resolution - Row 1 to 2 of 2
Command ===>                                             Scroll ===> PAGE

Commands: NEXT
Line commands:
 CH - Change  I - Interpret

Pending changes exist for table          JOHNSON.EMP
Apply virtual changes . . .          (Apply, Supersede, Ignore)

Sel Owner    Name               Statement
    *        *                  *
--- -------- ------------------ ----------------------------------------------->
    JOHNSON  EMP_CH2            ADMIN ALTER TABLE "JOHNSON"."EMP"   INSERT "MO
    JOHNSON  EMP_CH3            ADMIN ALTER TABLE "JOHNSON"."EMP"    ALTER COL
****************************** END OF DB2 DATA ******************************
```

Figure 452. Pending Changes - Conflict Resolution panel (ADB2CCR)

When you apply the pending changes, you define your new changes based on a virtual representation of the objects as if the pending changes were performed.

When you supersede the pending changes, you define the new changes without taking into account the effect of any pending changes; the new change becomes a prerequisite change for the pending changes, and any pending changes that are in ANALYZED status are set to DEFINED status.

When you ignore the pending changes, the new change you define does not become a prerequisite change for the pending changes. Any pending changes that are in ANALYZED status are not set to DEFINED status.

4. Fill in the Name field and any other options that you want to specify on the Register Options panel, and issue the CONTINUE command. When you make changes through ALT, and choose apply virtual changes, the **Replace existing change** field is not editable.

The following figure shows an example of the Register Options panel:

```
DB2 Admin ------------------- CM - Register Options ------------------- 21:36
Option ===>

Commands: CONTINUE                                      DB2 System: DB2X
                                                        DB2 SQL ID: JOHNSON
Specify the following values to register a change:

Owner . . . . . . . . JOHNSON  >  (Optional, default is JOHNSON)
Name  . . . . . . . . EMP_CH4                                          >
Comment . . . . . . . Increase the length of WORKDEPT                 >

Replace existing change . .                 ('/' to replace, Default is BLANK)

Specify the owner and name values to use for this change (? to lookup):
                   Owner        Name
Ignore  . . . . . . .            >                                    >
Mask  . . . . . . . .            >                                    >
```

Figure 453. Register Options panel (ADB2CRO)

Example 2: Registering a change that is created from screen input

This example shows how to register a change when you are prompted on the Register panel to specify whether to register the change as a normal or a fast change (emergency or immediate) and how any pending changes should be resolved. This example assumes that there are pending changes to the affected objects.

About this task

To register the change:

Procedure

1. Specify option 2.1 from the DB2 Admin main menu to display the Execute SQL Statements from Screen Input panel.
2. Enter the SQL statements that you want to run and press Enter.
3. If Change Management is optional for your SQL ID, specify YES when you are prompted whether to use Change Management in the Change Management Prompt pop-up panel that is displayed.
4. On the Register Change panel that is displayed, specify how the change should be registered, and press Enter.

The following figure shows an example of the Register Change panel:

```
   DB2 Admin ----------------- DB2X CM - Register Change ----------------- 21:36
   Option ===>  N

    C - Cancel
    E - Register and run as an emergency change
    N - Register as a normal change, pending changes become prereqs
    S - Register as a normal change, supersede pending changes
    G - Register as a normal change, ignore pending changes
    D - Display pending changes to the same object(s)

   For option E enter the following information for the change:
    Owner  . . . . .           > (Optional, default is JOHNSON)
    Name . . . . . .                     >
    Comment  . . . .                                                       >

   Statement that is about to be executed (first 28 lines)
   CREATE TABLESPACE HRTS1 IN HRB1


   +----------------------------------------------------------------------+
   | There are pending changes related to the objects you are modifying.  |
   | Use the "Display pending changes" option to see the pending changes. |
   +----------------------------------------------------------------------+
```

Figure 454. Example of Register Change panel (ADB2CMRG) when there are pending changes

> **Tip:** Use option D to review the pending changes to help you make the appropriate register decision for your change.

If you register the change as an emergency change or as a normal change that should supersede the pending changes, any pending changes that are in ANALYZED status are set to DEFINED status. They will need to be analyzed again.

If you register the change as a normal change and ignore the pending changes, any pending changes that are in ANALYZED status are not set to DEFINED status.

The following figure shows an example of the Register Change panel had there been no pending changes for the affected objects:

```
   DB2 Admin ----------------- DSN8 CM - Register Change ----------------- 21:36
   Option ===>

    C - Cancel
    I - Register and run as an immediate change
    N - Register as a normal change




   For option I enter the following information for the change:
    Owner  . . . . .           > (Optional, default is TONELLO)
    Name . . . . . .                     >
    Comment  . . . .                                                       >

   Statement that is about to be executed (first 28 lines)
   CREATE TABLESPACE HRTS1 IN HRB1
```

Figure 455. Example of Register Change panel (ADB2CMRG) when there are no pending changes

If you specify E or I on this panel to register the change as an emergency change (pending changes exist) or an immediate change (pending changes do not exist), you must specify an owner and a name for the change. An emergency or immediate change is run immediately.

5. Fill in the fields on the Register Options panel, and issue the CONTINUE command.

   The following figure shows an example of the Register Options panel:

```
DB2 Admin -------------------- CM - Register Options -------------------- 21:38
Option ===>

Commands: CONTINUE                                       DB2 System: DB2X
                                                         DB2 SQL ID: JOHNSON
Specify the following values to register a change:

Owner . . . . . . . . JOHNSON  >  (Optional, default is JOHNSON)
Name  . . . . . . . .                                                    >
Comment . . . . . . .                                                    >

Replace existing change . .            ('/' to replace, Default is BLANK)

Specify the owner and name values to use for this change (? to lookup):
                      Owner         Name
Ignore  . . . . . . .           >                              >
Mask  . . . . . . . .           >                              >
```

Figure 456. Register Options panel (ADB2CRO)

Analyzing a change

When you analyze a change, you run a job that creates a work statement list (WSL) that will be used in the run process to apply the changes.

About this task

A change must be in DEFINED or ANALYZED status to be analyzed.

To analyze a change:

Procedure

1. Display the change to be analyzed by selecting option 1 on the Change Management panel, and then select option 1 on the Manage Changes panel.
2. Issue the AN line command for the change that you want to analyze.
3. Fill in the fields on the Generate Analyze Job panel and press Enter. Specify the following information:
   - The base version method that DB2 Admin should use for the compare to generate the changes.

     If you specify U (User-defined), you are prompted to specify the version scope to use. If you specify E (Existing), you are prompted to specify the base version to use.
   - Whether to change reporting options before submitting the analyze job.

     If you specify YES, you are prompted to specify the reporting options to use.
   - Data set information for the WSL that is created and for the generated jobs.

     The value of the change tag type option determines the PDS member names.

     Tip: Keep the WSLs that are generated for changes made through change management separate from the other WSLs. Do not mix them in the same data set. Also, make the data set names for the WSLs and for the JCL unique enough so that members for different Change Management databases are not put in the same data set.
   - Job options.

If you choose to have DB2 Admin automatically generate a change that will recover the current change, you are prompted to register the recover change. When you choose to have a recover change generated, you must specify whether to have the original data or the existing data in the table recovered. Original data is the data that exists just before the original change is run. Existing data is the data that exists in the table just before the original change is recovered. Original data can be recovered only for objects that are dropped as part of the original change; referential integrity is not considered. Original data cannot be recovered for changes that are made using an SQL ALTER or RENAME statement.

- Optional utility job steps.
- Whether to use active templates.

    If you specify `Yes`, templates are generated for the non-utility data sets using the template definitions that are defined for Object Comparison Tool. If you specify `No`, the defaults for Prefix for data sets apply. If the Take an image copy or Run REORG options are Yes, the utility templates are used.

The following figure shows an example of the Generate Analyze Job panel:

```
ADB2C11A ----------------- Generate Analyze Job ----------------------- 21:45
Command ===>

Specify the following for Analyze:
                                                        More:     +
Base version method . . . . .         (Auto, User, or Existing)

Change reporting options  . . NO       (Yes/No)

Required data set information:
    PDS for WSL . . . . . . . . DSNA.RUN.WSL
    PDS for analyze job   . . . DSNA.ANALYZE.JCL
    Prefix for data sets  . . . JOHNSON
    Existing data set action  . CONDITIONAL (Conditional, Prompt, Replace)
    Change tag type . . . . . . ID         (ID, Name, Owner)

 Options:
    Run SQLID . . . . . . . . .           (Blank, a SQLID, or <NONE>
    Object Grantor  . . . . .             (Blank or a SQLID)
    Validate WSL  . . . . . . : NO        (Yes/No)
    Use utility options . . . . NO        (Yes/No)
    Generate templates  . . . . NO        (Yes/No)
    Build JCL to run WSL  . . . NO        (Yes/No)
    Generate a recover change . YES       (Yes/No)
      Data to recover . . . . . E         (Original or Existing)
      PDS for recover WSL . . . DSNA.RECOVER.WSL
      PDS for recover job . . . DSNA.RECOVER.JCL
    Authorization Switch ID . .           (SQLID to connect, <SQLID>, or blank)
    SECADM Authorization ID . .           (SQLID to connect or blank)

    Stop on conversion error. .           (Yes/No)
    Content of apply job(s) . . ALL       (All, DDL)
    Unload method . . . . . . . P         (Unload, Parallel unload, HPU)
    Use DEFER YES . . . . . . . YES       (Yes/No)
    Allow rotate parts  . . . . NO        (Yes/No)
    Retain GENERATED ALWAYS:
      For ROWID . . . . . . . . NO        (Yes/No)
      For ROW CHANGE TIMESTAMP. NO        (Yes/No)
    IDENTITY START value  . . . C         (Original, Computed)
    SEQUENCE RESTART value  . . C         (Original, Computed)
    Disable REORG optimization  YES       (Yes/No)

  Optional jobs after Reload or Alter:
    Run CHECK DATA  . . . . . NO        (Yes/No)
    Take an image copy  . . . N         (after: Reload/Alter/Both/None)
    Run REORG/REBUILD . . . . N         (Mandatory, All relevant, None)
    Run RUNSTATS  . . . . . . N         (after: Reload/Alter/Both/None)
    Run REBIND  . . . . . . . NO        (Yes/No)
BP - Change batch job parameters
TU - Specify TEMPLATE usage
UO - Change utility options
CO - Change options common to change functions
```

Figure 457. Generate Analyze Job panel (ADB2C11A)

Depending on the values that you specify on the panel, you might be
prompted for additional information before the jobs to perform the analysis are
generated and before an ISPF Edit session is displayed.

4. If the change that you are analyzing has already been analyzed (that is, the
   change is in ANALYZED status, specify whether to continue with or to cancel
   the analyze request when you are prompted. The warning prompt indicates
   that the change will be put back in DEFINED status before the new analyze job
   is created if you continue.
5. Edit and submit the generated job. When the job completes successfully, the
   change is placed in ANALYZED status.

   If you requested that a recover change be generated, the recover change is
   created and is also placed in ANALYZED status. In addition, a delta version for
   the recover change is created.

6. Press PF3 to return to the Changes panel to verify that the status of the change is ANALYZED. If you requested that a recover change be generated, you can verify that it is included in the list of changes.

   Tip: If you return to the Changes panel before the submitted job completes, you can enter the REF primary command after the job completes to see the refreshed status of the change.

What to do next

If the job does not complete successfully, check the error messages in the job output. Correct any errors and then reanalyze the change by issuing the AN command.

Base version method
During the analysis of a change, DB2 Admin needs to know the current state of the objects that are being changed.

DB2 Admin can get this information from an existing version that was created earlier or extract the information from the DB2 catalog.

When the information is extracted from the DB2 catalog, DB2 Admin either extracts it based on a user-defined scope or based on the objects that are being changed.

The base version method that you choose depends on your installation's needs. Your shop might prefer to create a new snapshot (base version) after every change to use as a backup and also as the base version for new changes. When the next change needs to be analyzed, you can specify to have the existing version used and avoid extracting the object definitions from the DB2 catalog to get the current status. Processing time is saved when you do not have to extract the objects from the catalog.

Other shops might want to work on one application at a time. A scope can be defined that includes all of the objects in the application (for example, one or more databases) and always use this scope as the base when analyzing a change.

Some shops might not want to use existing base versions or user-defined scopes and choose to have the base automatically generated from the DB2 catalog when analyzing a change.

Running a change
When you run a change, the work statement list (WSL) that was created during the analyze process is run.

About this task

A change must be in ANALYZED status to be run. If you plan to have a base version of the objects created after the change, a version scope that defines the set of objects to be included in the base version must exist.

To run a change:

Procedure

1. Display the change to be run by selecting option 1 on the Change Management panel, and then select option 1 on the Manage Changes panel.

2. Issue the RN line command for the change that you want to run. If the change has prerequisite changes, DB2 Admin will issue a message that prompts you to run the prerequisite changes first.

3. Fill in the fields on the Run a Change panel and press Enter. Specify the following information:

   - **Data set information**: Data set information for the generated jobs.

   - **Change reporting options**: Whether to change the Object Compare reporting options for the runtime analyze.

     Changes, such as those that are not made through Change Management, might have occurred to the DB2 catalog since the WSL that was generated during the analyze process was created. That WSL might now conflict with or undo those changes. To minimize the possibility of run-time errors, you can verify the WSL by generating a new WSL and having it compared to the WSL that was generated during the normal analyze process.

     The new run-time WSL is generated for the change, and its recover change if one exists, based on the current DB2 catalog and using the automatic base version method. The run-time WSL files are compared with the WSL files that were created during the normal analyze process. If the DDL and DCL content are the same, the run job continues and the WSL that was generated during the normal analyze is run to apply the changes. If the DDL or DCL content are different, the run job stops with a return code of 8, and the change is not run. A message is issued to indicate that the WSLs did compare equally. If the WSLs mismatch, go the Changes panel and issue the AN line command to analyze the change again to resolve the differences before trying to run the change again.

   - **Generate base version before run**: Whether to generate a new base version just before the change is implemented. The following options are available:
     - **No**: A base version is not created before the change is implemented. The objects in the base version are automatically determined by the product, based on the objects being changed.
     - **Auto**: A base version is created before the change is implemented. The objects in the base version are the ones listed in the user-specified version scope.
     - **User**: A base version is created before the change is implemented. The objects in the base version are the ones listed in the user-specified version scope.

     Note: If Auto or User is chosen, the Specify Base Version Options panel (ADB2CEX3) appears and collects the name of the new base version. The name of a version scope is also collected if User was chosen.

   - **Generate base version after run**: Whether to generate a new base version just after the change is implemented. The following options are available:
     - **No**: A base version is not created after the change completes.
     - **Auto**: A base version is created after the change completes. The objects in the base version are automatically determined by the product based on the objects being changed.
     - **User**: A base version is created after the change completes. The objects in the base version are the ones listed in the user specified version scope.

Note: If Auto or User is chosen, the Specify Base Version Options panel (ADB2CEX3) appears and collects the name of the new base version. The name of a version scope is also collected if User was chosen.

The following figure shows an example of the Run a Change panel:

```
ADB2CEX1  -------------------- CM - Run a Change ----------------------- 15:33
Command ===>


Change . . . : VNDR12.S30568

Specify the following for run change:

Data set information:
  PDS for run job  . . . . . . DSNA.RUN.JCL
  Prefix for data sets . . . . VNDR12
  Existing data set action . . C          (Conditional, Prompt, Replace)
  Change tag type  . . . . . . ID         (ID, Name, Owner)


Change reporting options . . . . . . NO   (Yes/No)
Generate base version before run . . NO   (No,Auto,User)
Generate base version after run  . . NO   (No,Auto,User)

Do runtime analyze . . . . . . . . .      (Yes/No)
******************************** END OF DB2 DATA ********************************
```

Figure 458. Run a Change panel (ADB2CEX1)

The following figure shows an example of the Specify Base Version Options panel:

```
DB2 Admin ----------------- CM - Specify Base Version Options ----------------------
Command ===>

Commands: NEXT

Change . . . : VNDR12.VN236692012-03-06-09.45.53.415055

Specify the following for the base versions:

Base version before run:
  Scope Information:
    Owner . . . . . . . .          >           (? to lookup)
    Name . . . . . . . .                         > (? to lookup)

  Version Information:
    Owner . . . . . . . .          >           (? to lookup)
    Name . . . . . . . .                         > (? to lookup)

Base version after run:
  Scope Information: the object list will be automatically determined.
    Owner . . . . . . . :          >           (? to lookup)
    Name . . . . . . . . :                        > (? to lookup)

  Version Information:
    Owner . . . . . . . .          >           (? to lookup)
    Name . . . . . . . .                         > (? to lookup)
******************************** END OF DB2 DATA ********************************
```

Figure 459. Specify Base Version Options panel (ADB2CEX3)

Tip: When a version scope is used for the base version and you create a new base version and the change is for an object outside of the current version scope, ensure that you update the definition of the version scope. You want to update the version scope to include all objects so that any subsequent changes for which you create a new base version include all the objects. For example, if

the version scope includes database DB01 and the change is to add a second database DB02, change the definition of the version scope to include database DB02.

4. Edit and submit the generated job. The change is placed in COMPLETE status.

   When you run a change, the run job reanalyzes the change and creates a second WSL. This second WSL is compared with the WSL that was generated during the normal analyze process. If the DDL and DCL content are the same, the run job continues and the WSL that was generated during the normal analyze is run to apply the changes. If the DDL or DCL content are different, the run job stops with a return code of 8, and the change is not run. A message is issued to indicate that the WSLs did compare equally. If the WSLs mismatch, go the Changes panel and issue the AN line command to analyze the change again to resolve the differences before trying to run the change again.

5. Press PF3 to return to the Changes panel to verify that the status of the change is COMPLETE.

   Tip: If you return to the Changes panel before the submitted job completes, you can enter the REF primary command after the job completes to see the refreshed status of the change.

What to do next

If the job fails (the status of the job does not change to COMPLETE), the action to take depends upon the status in which the change is left:

- If the status is ANALYZED, check the job output. If a message indicates that the run-time WSL did not match the WSL that was generated during the analyze process, return to the Changes panel and issue the AN line command to reanalyze the change. Then, run the change again.
- If the status is RUNNING, check the job output. Determine the cause of the failure and make any necessary corrections. Then, return to the Changes panel, issue the ER line command to edit the run job, and resubmit it. When you submit the run job, the job is restarted at the appropriate step.

  When you issue the ER line command, the JCL for the run job is placed in edit mode. Before the job is displayed in edit mode, a RESTART parameter is automatically added to the job card to restart the job at the step that runs ADBTEP2 so that you do not have to determine the step name where the job should be restarted. In addition, if the RESTART parameter for ADBTEP2 was changed to RESTART(NO) by using the ER line command during an earlier edit session, the parameter is automatically changed to RESTART(YES) because ADBTEP2 must be restarted with the parameter RESTART(YES). If the parameter is missing, ADBTEP2 assumes a YES value.

  These automatic changes and any edit changes that you make are saved to the JCL data set so that you do not need to re-enter the changes for a subsequent ER line command for the job.

  When you submit the run job, the job is restarted at the appropriate step.

  Important: Any user can use the ER line command to edit and resubmit a change in RUNNING status. The user who originally ran the change is not required to resubmit the job. The restart record in the checkpoint table for the change retains the userid of the original submitter. DB2 Admin locates the record by using CHANGEID parameter. The RN and ER line commands automatically include the CHANGEID parameter when the run job is built so that you do not have to manage this process.

Exporting changes

You can selectively export multiple changes made in one environment and distribute those changes to multiple external environments.

About this task

You can promote changes made in one environment to different environments. You can create a list of the changes whose statements are to be promoted. The changes can be arranged in any desired sequence, and you can select which changes to promote.

When the set is complete, you can extract all of the change statements to a single file. The file might then be imported in a different environment. As in the current promote process, a change type of COMPARE is created and marked COMPLETE when the promote has ended. The statements are used by the existing import function to carry out the change in the target environment.

When exporting changes:
- If the exported SQL statements affect objects for which pending changes exist, then the system determines whether the change becomes a prerequisite change for those pending changes.
- You can create a single change by exporting multiple files at the same time. All types can be part of the same export.
- When you export SQL statements into a change, the version of DB2 that is on the system must support the SQL statements that you are exporting.

The following steps described how to specify changes to export from panel ADBPC15. You can also issue the **EX** line command on the ADB2C11 panel to select changes to export individually. When you enter the **EX** command, you then can view all selected changes on panel ADBPC15.

You can use the search criteria fields in panel ADB2C1, to qualify changes. The filtered changes and prerequisites changes are displayed on panel ADBPC15.

To export a change:

Procedure

1. Select option 1 on the Change Management panel to display the Manage Changes (ADB2C1) panel.
2. Select option 5, Export Changes. Panel ADBPC15 is displayed and lists all changes. By default, all changes are marked as INCLUDE. You can issue the XC line command to exclude individual changes from the list. On the ADBPC15 panel, you can issue the **XC** line command to exclude a change or the **IC** line command to include a change. The following figure shows the Export Changes panel:

```
ADBPC15 n --------------- CM - Export Changes --------------------------
Commands: NEXT   ADD   COMMENT   EXPOPT
Line commands:
 I - Interpret   IC - Include Change   XC - eXclude Change
                                                                Operation
Sel       ID Owner    Name                          Type    Status   Type
          * *         *                             *       *        *
--- -------- -------- ------------------------------> ------- -------- ---------
      1066 RAXESHP  D26985                        CHANGE  DEFINED  INCLUDE
      3883 J148286  AUTO:2013-09-18-09.54.12.50428 CHANGE  ANALYZED INCLUDE
         1 SCHAUFU  D24583A                       CHANGE  COMPLETE INCLUDE
      1064 VNDLRC   DT26897.CHANGE00.02           CHANGE  COMPLETE INCLUDE
      1061 VNDRG    D27018 A2SMPETEST             CHANGE  ANALYZED INCLUDE
      1060 VNDLRC   DT27024.CHANGE.01             CHANGE  ANALYZED INCLUDE
      1059 VNDLRC   DT27024.CHANGE.00             CHANGE  COMPLETE INCLUDE
        22 VNDEJB   EBX2                          CHANGE  DEFINED  INCLUDE
         4 RAXESHP  TST1                          CHANGE  DEFINED  INCLUDE
         3 VNDEJB   DSFA                          CHANGE  DEFINED  INCLUDE
      1053 XHLI     CHG00002                      CHANGE  ANALYZED INCLUDE
      1052 XHLI     CHG00001                      CHANGE  DEFINED  INCLUDE
Command ===>                                              Scroll ===> PAGE
 F1=HELP     F2=SPLIT     F3=END      F4=RETURN   F5=RFIND    F6=RCHANGE
 F7=UP       F8=DOWN      F9=SWAP     F10=LEFT    F11=RIGHT   F12=RETRIEVE
```

Figure 460. Export Changes panel (ADBPC15)

3. Optional: Issue the EXPOPT command to specify batch mode or specify a
   mask. You can also use this command to specify that you want to show the
   options panel prior to each use of the Export Changes panel.

```
ADBPC15O ---------------------- DSNA Export Options --------------------- 10:52
Option ===>

Please specify the following Export options:

Export changes in batch . . . . . . . YES (Yes/No)

Enter mask details (optional):
  Mask Table Entry:
    Owner . . . .      >                 (? to look up)
    Name  . . . .                  >     (? to look up)
  Data Set:
    Mask DSN  . .
  Options:
    Edit Mask . .                        (Yes/No)

Show this panel prior to each use . . YES (Yes/No)
```

Figure 461. Export Options panel (ADBPC15O)

Exporting a change is a two-phase process in which DB2 Admin determines if
there are any pending changes for the objects and then registers the exported
change. The processing modes are:

TSO   Perform the processing in the foreground (TSO). This is the default.

Batch Perform the processing in background (batch)

If you specify a mask, the mask is applied to the changes that you selected on
the Export Changes panel (ADBPC15). The exported data set will have the
specified mask applied.

4. To process the export, issue the CONTINUE command.

Panel ADBPVERD is displayed after issuing the CONTINUE command. Specify
parameters for the dataset that will contain the final list of exported changes.
This dataset can be used as a changes file to be imported later using option 4
on panel ADB2C1.

```
  ADBPVERD  -------------- Specify Data Set / Member Information -------------

  Data Set Name  . . EXPORTED.CHANGES
  *Member Name . . .

  *Volume serial . . . . :              (Blank for system default volume)
  Device type  . . . . . . SYSALLDA    (Generic unit)
  Space units  . . . . . . TRACKS      (TRKS or CYLS)
  Primary quantity . . . . 1           (In above units)
  Secondary quantity . . . 1           (In above units)
  *Directory blocks  . . . 0           (Zero for sequential data set)     *
  *Record format . . . . : F           (F or V)
  *Record length . . . . : 80          F80
  *Block size  . . . . . .
  *Data set name type  . .              (LIBRARY, PDS or blank)
  (* Specifying LIBRARY may override zero directory block)


   F1=HELP    F2=SPLIT    F3=END     F4=RETURN   F5=RFIND    F6=RCHANGE
   F7=UP      F8=DOWN     F9=SWAP    F10=LEFT    F11=RIGHT
```

Figure 462. Specify Data Set / Member Information panel (ADBPVERD)

Exporting multiple data sets into a single change should be carefully planned.
Export cannot check whether the changes in the specified sequence will
logically work as desired. The changes will be imported into the change
individually in the sequence they are specified, and you must ensure that any
change in the list logically has all preceding changes as prerequisites.

Results

You can now display your exported changes on the Export Changes panel.

Multi-target changes

You can register a change to any catalog object on one system and import the
change on multiple target systems. Changes are prepared on a central system and
then applied to one or more target systems.

Before you begin

Admin Tool calls stored procedure ADBCRSP to update the change management
database for multi-target changes. You should configure the WLM address space so
that it has access to load modules ADBCRSP, ADB3000, and ADB9000 by copying
these modules to a library defined in the STEPLIB concatenation for the WLM
address space. The following example uses the ADMIN.WLM.LOAD address
space:

```
000024 //STEPLIB  DD   DSN=ADMIN.WLM.LOAD,DISP=SHR
000025//          DD  DSN=USER.TESTLIB,DISP=SHR
000026 //         DD  DSN=DB2A.UTLIB,DISP=SHR
000027 //         DD  DSN=DB2A.TESTLIB,DISP=SHR
000028 //         DD  DSN=DB2A.SDSNLOAD,DISP=SHR
000029 //         DD  DSN=DB2A.SDSNLOD2,DISP=SHR
000030 //         DD  DSN=CEEA.SCEERUN,DISP=SHR
```

About this task

The following topics show you how you might deploy a change on multiple
targets.

Setting up the targets
You can set up all the targets that you want to deploy changes to through the
Change Management panel.

About this task

Procedure

1. Select option 9, Manage targets on the Change Management panel. The CM - Manage Targets panel is displayed, as shown in the following figure:

```
ADBPC9 in ------------------ CM - Manage Targets  --------------------- 16:06
Option ===>


  1 - Display targets                              DB2 System: DSNB
  2 - Display target groups                        DB2 SQL ID: WBELIS
  3 - Insert a target


Enter display selection criteria.  Settings: LIKE operator;  Criteria not saved
Target name  . . . .                    >      Group name . .          >
Location name  . . .                    >      Created by . .          >
                                               Altered by . .          >
```

Figure 463. CM - manage Targets (ADBPC9)

On the Manage Targets panel, you can display targets or create a target. When you use option 1 or 2, you can qualify the search by using the additional search criteria fields at the bottom of the screen.

2. Select option 3, Insert a target on the Manage Targets panel. The Insert a Target panel is displayed, as shown in the following figure:

```
ADBPC911  --------------- CM - Insert a Target ------------------------- 15:50
Command ===>


Type new values and press Enter.

*Name  . . . . . . . . . TARGET A            > (? to lookup)
*DB2 location  . . . . . DSNA                > (? to lookup)
Comment  . . . . . . . .                     >
*Communication method  . DRDA                (DRDA or File)
Mask owner at target . . SYSADM              >
Mask name at target  . . MASK A              >
```

Figure 464. CM - Insert a target panel (ADBPC911)

The following fields are displayed on this panel:

Name   The name given for the target.

DB2 location
       The location of the remote server.

Comment
       An optional field you use to enter a comment to describe the target.

Communication method
       Specify the method used to register changes to this target:

   DRDA
           Use DRDA when registering changes to this target.

   FILE   Use the file method to register changes to this target. Note that a file is written with information for all targets regardless of which method is specified.

Mask owner at target
> Specify the owner of the default mask that exists at the target location.

Mask name at target
> Specify the name of the default mask that exists at the target location.

3. Set up the new target by specifying the details on the CM - Insert a Target panel and then press Enter.

   The target is inserted.

4. To add another target, repeat 3 until all targets are configured.

Displaying targets

You can manage target profile definitions and specify selection criteria for displaying a list of target profiles.

About this task

You can create a multi-target change in which changes you make on a central system are propagated to one or more targets. A target is a DB2 subsystem where you wish to apply the change made on the central system.

Procedure

1. Select option CM, Change Management on the DB2 Administration Menu.
2. Select option 9, Manage targets on the Change Management panel.
3. Select option 1, Display targets on the CM - Manage Targets panel. The following figure shows the CM - Targets panel.

```
ADBPC91 n ------------------ CM - Targets -------------------- Row 1 to 2 of 2
Command ===>                                               Scroll ===> CSR

Line commands:
 U - Update   DEL - Delete   INS - Insert  I - Interpret

Sel Name               DB2 Location       Comment
    *                  *                  *
--- -----------------  -----------------  ------------------------------------
    DB210CONV          DBAD               DB2 10 CONVERSION
    DB210NFM           DSNA               DB2 10 New function Mode
***************************** END OF DB2 DATA ******************************
```

Figure 465. Manage Targets panel (ADBPC91)

4. Select one of the following line commands to work with the target.

   U   Update the current target entry using the Insert a Target panel (ADBPC911).

   DEL   Delete the current target entry.

   INS   Insert a target panel (ADBPC911).

   I   Provide an interpretation of the target. This option displays the name, DB2 location, comment, communication method, the mask name and owner at the target, the ID of the person who created the target, and the date it was last altered.

Displaying target groups

A target group is an optional entity that represents a set of target environments. You create a group name and select the targets that comprise the group. You can create or display target groups.

About this task

A target can be defined in one or more groups. Groups can be redefined as needed.

To display target groups:

Procedure

1. Select option CM, Change Management on the DB2 Administration Menu.
2. Select option 9, Manage targets on the Change Management panel.
3. Select option 2, Display target groups, on the CM - Manage Targets panel. If no target groups exist, panel ADBPC921 is displayed, allowing you to insert a group. If a target group exists, panel ADBPC92 is displayed, as shown in the following figure:

```
ADBPC92 DTEST ------------------- CM - Target Groups -------------- Row 1 of 2
Command ===>                                            Scroll ===> PAGE

Line commands:
 INS - Insert  T - Targets

Sel Group Name                                          Targets
    *                                                   *
--- -------------------------------------------- -----------
    PROD                                                1
    TEST                                                2
***************************** END OF DB2 DATA ******************************
```

Figure 466. Manage Targets panel (ADBPC92)

4. Select one of the following line commands to work with the target group.
   - Selecting the INS line command displays panel ADBPC921, as shown in the following figure:

```
ADBPC921 DTEST ------------ CM - Insert a Group ------------------------ 08:27
Command ===>


Group name . .                                    > (? to lookup)
Target name  .                                    > (? to lookup)
```

Figure 467. Insert a Group panel (ADBPC921)

   This panel allows you to insert a target group and target location entry. Enter the Group name to indicate the name of the target group, and the Target name to indicate the target name to include in the group.
   - Selecting the T command displays panel ADBPC92T, as shown in the following figure:

```
ADBPC92T  -------------------- CM - Targets in a Group ------------ Row 1 of 1
Command ===>                                            Scroll ===> PAGE

Line commands:
 A - Add target  R - Remove target  S - Show target

Sel Target Name                       Group Name
    *                                  *
--- ---------------------------------- ----------------------------------------
    DSNA                               PRODUCTION
***************************** END OF DB2 DATA *******************************
```

Figure 468. Targets in a Group panel (ADBPC92T)

> The S line command displays panel ADBPC91. The R line command removes
> the target from the group. If it is the last target in the group, the group is
> removed.

Registering a multi-target change

You can register and track changes on multiple target systems.

Before you begin

Change Management must be enabled on the system and be either optional or
required for your SQL ID. You enable Change Management by customizing the
DB2 Admin Tool. For more information, see the "Customizing DB2 Admin" chapter
in the *DB2 Administration Tool User's Guide*.

If the option to create a multi-target change is shown on the Register Options
panel (ADB2CRO), then the change can be registered on multiple target systems.
The steps that follow assume your system is configured to create a multi-target
change.

You can register the change on multiple target locations.

About this task

To register a multi-target change:

Procedure

1. Specify Yes in the Multi-target Change field on the CM - Register Options panel
   and then issue the CONTINUE command. The following figure shows an
   example of the CM - Register Options panel:

```
ADB2CRO n ------------------ CM - Register Options ------------------ 17:30
Command ===>

Commands: CONTINUE                                    DB2 System: DSNX
                                                      DB2 SQL ID: VNDR1
Specify the following values to register a change:

Owner . . . . . . . . . VNDR1          > (Optional, Default is VNDR1)
Name  . . . . . . . . . TESTCHG1                                        >
Comment . . . . . . .                                                  >
Multi-target Change . YES              (Yes/No, Default is NO)
  Target Name . . . . TESTTEST3        > (Optional, ? to lookup)
  Group name  . . . .                  > (Optional, ? to lookup)

Replace existing change . .            ('/' to replace, Default is BLANK)

Specify the owner and name values to use for this change (? to lookup):
                    Owner        Name
Ignore  . . . . . . .         >                                  >
Mask  . . . . . . . .         >                                  >
```

Figure 469. CM - Register Options Panel (ADB2CRO)

2. Select the target names you want to register on the ADBPCMT panel then select NEXT. The following figure shows an example of the CM - Associate Targets panel:

```
 ADBPCMT n --------------- CM - Associate Targets ------------- Row 1 to 1 of 1

Details for multi-target change: VNDR1.S28479-C1       DB2 System: DSNX
                                                       DB2 SQL ID: VNDR1
Commands: NEXT
Line commands:
  U - Update  I - Interpret

     Target                Change    Change
Sel Name     DB2 Location  Owner     name                   Status
     *        *            *         *                       *
--- ------->  ------------> --------- ---------------------- -------------------
 I  TESTCHG1 DSNA          ATCOWN    ATCNAM                 NEW
****************************** END OF DB2 DATA ******************************
```

Figure 470. CM - Associate Targets Panel (ADBPCMT)

Note: If no targets exist, panel ADBPC911 displays to allow you to insert targets.

3. Specify the action to take for any pending changes to the objects on the target system that are affected by this change:

Cancel
> Do not register the change if there are pending changes.

Prereq Make the pending changes for the affected objects prerequisite changes for this change.

Supersede
> Make this change a prerequisite change for the pending changes.

```
.----------------------------------------------------------------------.
|                                                                      |
| GOC5RM  ----------------- Specify Register Mode ----------------- 17:35 |
|                                                                      |
| Pending changes action . . SUPERSEDE (Cancel, Prereq, Supersede)     |
|                                                                      |
|                                                                      |
|                                                                      |
|   F1=HELP      F2=SPLIT     F3=END       F4=expand    F5=RFIND    F6=RCHANGE |
|   F7=UP        F8=DOWN      F9=SWAP      F10=LEFT      F11=RIGHT         |
'----------------------------------------------------------------------'
```

Figure 471. Specify Register Mode Panel (GOC5RM)

If successful, the output indicates Register Successful and the changes are
registered on the specified targets. The following is an example report of a
multi-target change summary:

```
ADB2CID - Multi-Target Change Summary

Multi-target change id:        3747

Target    Owner    Name              Status
--------- -------- ----------------- -----------------------------------
          B148286  A                 ADB9400I:The change was registered successfully, Changeid: 3957
          C148286  B                 ADB9400I:The change was registered successfully, Changeid: 3958

ADB2CID - Multi-Target Change End of Summary
```

Figure 472. Multi-Target Change (ADB2CID)

Importing multi-target changes

You can view all of the change statements in a target file before they are imported
to a target system. A *target file* contains only one change, but the change can have
multiple statements.

About this task

Importing multi-target changes is similar to importing changes to a single target.
An additional panel is displayed when importing multi-target changes.

When you import changes to the local target, you can import the change
statements that are contained in the file to a (single) local target DB2 subsystem.

To import a multi-target change:

Procedure

1. Select option 1 on the Change Management panel to display the Manage
   Changes panel.
2. Select option 4 to import changes.
3. Specify the name of the data set that contains the multi-target change
   statements (see "Multi-target changes" on page 585). Only a single data set
   containing multi-target change content can be imported at one time. The
   following figure shows the Import Changes panel:

```
ADB2C14 DTEST -------------- CM - Import Changes ---------------- Row 1 of 2
Command ===> continue

Commands : CONTINUE  RESET

Input data set information:                              DB2 System: DSNA
  Data set name . TEST2.MTC
    Member  . . .         (member name or pattern if partitioned)

Line Commands :
M - Move  A - After  B - Browse  D - Delete

Select Seq Data set name                                        Oper.
------ --- ------------------------------------------------------    ------

       1 J148286.TEST2.MTC
****************************** END OF DB2 DATA ******************************
```

Figure 473. Import Changes panel (ADB2C14)

To process the import, issue the CONTINUE command. To clear the list of data sets, issue RESET.

4. If you are importing a multi-target change the Import changes to the local target panel is displayed, as shown in the following figure:
   The target change fields are input fields, allowing you to override the contents

```
ADBPC14L DTEST ------- CM - Import changes to the local target ---- Row 1 of 2
Command ===>

Central change . . : J148286.MTC33

Commands: NEXT
Line Commands:  A - Add  D - Delete  R - Repeat

S Information  Owner              Name
- -----------  ------------------ ---------------------------------------------
* Change . . . J148286           > CHANGE1                                     >
    Mask . . .                   >                                             >
    Ignore . .                   >                                             >

    Comment  .                                                                >
****************************** END OF DB2 DATA ******************************
```

Figure 474. Import Changes to the local target panel (ADBPC14L)

of the multi-target change file.

5. You can use masking to affect different objects. A mask allows you to change the object names as they are read from the file, which allows you to affect a different set of objects on the target. You specify the mask name and owner, as in the following figure. Note that optional ignores can also be specified while registering the change; however, ignores will be applied while analyzing the change.

```
ADBPC14L DTEST ------- CM - Import changes to the local target ---- Row 1 of 1
Command ===>

Central change . . : J148286.MTC33

Commands: NEXT
Line Commands:  A - Add  D - Delete  R - Repeat

S Information  Owner                Name
- -----------  ------------------   --------------------------------------------
  Change . . . J148286          > CHANGE1                                    >
    Mask . . . J148286          > MASK_1                                     >
    Ignore . .                  >                                            >

    Comment  .                                                               >
  --------------------------------------------------------------------------------
****************************** END OF DB2 DATA ******************************
```

Figure 475. Import Changes to the local target panel (ADBPC14L)

6. Use the A (Add) or R (Repeat) line commands to specify additional changes. For example, two additional changes have been added in the following figure:

```
ADBPC14L DTEST ------- CM - Import changes to the local target ---- Row 1 of 3
Command ===>

Central change . . : J148286.MTC33

Commands: NEXT
Line Commands:  A - Add  D - Delete  R - Repeat

S Information  Owner                Name
- -----------  ------------------   --------------------------------------------
* Change . . . J148286          > CHANGE1                                    >
    Mask . . .                  >                                            >
    Ignore . .                  >                                            >

    Comment  .                                                               >
  --------------------------------------------------------------------------------
* Change . . . J148286          > CHANGE2                                    >
    Mask . . . J148286          > MASK_2                                     >
    Ignore . .                  >                                            >

    Comment  .                                                               >
  --------------------------------------------------------------------------------
* Change . . . J148286          > CHANGE3                                    >
    Mask . . . J148286          > MASK_3                                     >
    Ignore . .                  >                                            >

    Comment  .                                                               >
  --------------------------------------------------------------------------------
****************************** END OF DB2 DATA ******************************
```

Figure 476. Import Changes to the local target panel (ADBPC14L)

7. Issue the NEXT command. When entered, the NEXT command builds a batch job that registers the change(s) on the system. After submitting the batch job you can display your imported change on the Changes panel, analyze the change, and then run it.

Each change owner and name specified on the panel must be unique because the change statements in the file are imported to the same DB2 subsystem. Even if you use unique change owner/names, you should not have the same objects affected by the same change statements more than one time. You should specify different masks to affect changes to different objects.

You can determine whether a change has already been registered with the same multi-target change ID or the same mask as the one you are importing. If an existing change is identified, it will be identified with one of the following statuses:

Initial   The change will be restarted. Supersede and prerequisite decisions will be used.

Defined, Analyzed, Complete, or Running:
> The change will remain as it is currently defined. The input change is ignored.

Canceled
> The input change is registered.

The input change name and owner should not conflict with the existing canceled change.

Exporting multi-target information to a dataset on the target system

You can consolidate status updates on the target system for one or more multi-target central systems.

About this task

You can consolidate status updates for parameters, selection criteria and other related options to communicate the updates to a central system.

When the target system does not have DRDA connectivity to the central system, the continuous updates for the multi-target changes are not communicated to the central system. You can consolidate updates on target changes, including status and other information, into a data set which can eventually be processed on the central system to synchronize the central system with the target systems.

To export a multi-target change, you can use either a batch or online process:

Procedure

1. To export a multi-target change using batch:
   a. Access Change Management by using the CM option on the DB2 Administration Menu panel.
   b. Select option 1 on the Change Management panel to display the Manage Changes panel.
   c. Select option 6 on the CM - Manage Changes panel. The Specify Data Set / Member Information panel is displayed, as shown in the following figure.

```
ADBPVERD  -------------- Specify Data Set / Member Information -------------

Data Set Name  . . MTC.UPDATES
*Member Name . . .

*Volume serial . . . . :              (Blank for system default volume)
Device type  . . . . . . SYSALLDA     (Generic unit)
Space units  . . . . . . TRK          (TRKS or CYLS)
Primary quantity . . . . 1            (In above units)
Secondary quantity . . . 1            (In above units)
*Directory blocks  . . . 20           (Zero for sequential data set)     *
*Record format . . . . : F            (F, FB, V, or VB)
*Record length . . . . : 80           F80,FB80
*Block size  . . . . . .
*Data set name type  . .              (LIBRARY, PDS or blank)
(* Specifying LIBRARY may override zero directory block)
```

Figure 477. Specify Data Set / Member Information (ADBPVERD)

    d. Specify the name of the data set that is to contain the target information.

    e. If the dataset already exits, you are asked whether to replace the contents of the dataset or cancel. Select the appropriate option and press Enter. A new job is generated and displayed. The SYSIN parameters can be changed before submitting the job. The parameters that can be specified in the batch job are: ALTERAGE, MTCLOC, MTCIDS, and CHGIDS.

ALTERAGE
> Specifies the target changes that were altered during a specified period. Acceptable formats are: N YEAR(S), N MONTH(S)", N DAY(S), N MINUTE(S), or N SECOND(S). For example, ALTERAGE="1 MONTH".

CHGIDS
> Specifies a comma-separated list of target change ID values. For example: CHGIDS="1, 11, 40, 1001"

MTCIDS
> Specifies a comma-separated list of multi-target change ID values. For example, MTCIDS="10, 1000, 3100".

MTCLOC
> Specifies a list of multi-target central locations. For example, MTCLOCS="'DSNA','DSNB','DSNC'"

After the job is run successfully the output file will contain multi-target information, as shown in the following figure.

```
VIEW       VIJAYAK.MTC.UPDATES                           Columns 00001 00072
Command ===>                                             Scroll ===>; CSR
****** **************************** Top of Data ****************************
000001 <TARGETINFO VERSION="1">
000002 <MTCLOCATION>DSNA
000003 <TARGETLOCATION>DSNB
000004 <CHANGE>
000005 <MTCCHANGEID>2578</MTCCHANGEID>
000006 <OWNER>QMFADM</OWNER>
000007 <NAME>AUTO:2013-06-14-09.07.46.578784</NAME>
000008 <STATUS>INITIAL</STATUS>
000009 <MASK>
000010 <OWNER></OWNER>
000011 <NAME></NAME>
000012 </MASK>
000013 <CREATEDTS>2013-06-14-09.07.47.824553</CREATEDTS>
000014 <ALTEREDTS>2013-06-14-09.07.47.824553</ALTEREDTS>
000015 </CHANGE>
000016 <CHANGE>
000017 <MTCCHANGEID>2637</MTCCHANGEID>
000018 <OWNER>QMFADM</OWNER>
000019 <NAME>AUTO:2013-06-15-17.41.44.561870</NAME>
```

Figure 478. Example job to export changes to a dataset.

A report is also generated as shown in the following figure:

```
******************************** TOP OF DATA ********************************

Multi-target changes Report:
============================

< MTC Details:   > < Target Details:                          Altered
Location  ChangeID ChangeID  Owner    Change Name        Status     Timestamp
--------  -------- --------  --------  ------------------ ---------  ----------
DSNA          2578      226 QMFADM   AUTO:2013-06-14-09 INITIAL   2013-06-14

DSNA          2637      280 QMFADM   AUTO:2013-06-15-17 DEFINED   2013-06-15

DSNA          2674      292 QMFADM   AUTO:2013-06-18-14 DEFINED   2013-06-18

DSNB           259      260 VIJAYAK  AUTO:2013-06-15-16 INITIAL   2013-06-15

DSNB           261      262 VIJAYAK  AUTO:2013-06-15-16 INITIAL   2013-06-15

DSNB           263      264 VIJAYAK  AUTO:2013-06-15-16 INITIAL   2013-06-15

DSNB           265      266 VIJAYAK  AUTO:2013-06-15-16 INITIAL   2013-06-15

DSNB           267      268 VIJAYAK  AUTO:2013-06-15-16 INITIAL   2013-06-15

DSNB           269      270 VIJAYAK  AUTO:2013-06-15-16 INITIAL   2013-06-15

DSNB           271      272 VIJAYAK  AUTO:2013-06-15-16 INITIAL   2013-06-15

DSNB           274      275 VIJAYAK  AUTO:2013-06-15-17 DEFINED   2013-06-15

DSNB           277      278 VIJAYAK  AUTO:2013-06-15-17 DEFINED   2013-06-15

DSNB           285      286 VIJAYAK  AUTO:2013-06-15-18 ANALYZED  2013-06-15

DSNB           287      288 J148286  AUTO:2013-06-17-10 DEFINED   2013-06-18

******************************** BOTTOM OF DATA ********************************
```

Figure 479. Multi-target changes Report

If a search criteria resulted in no rows found, a warning is displayed and
the job ends with RC=8.

2. To export using online:

a. Access Change Management by using the CM option on the DB2 Administration Menu panel.

b. Select option 1 on the Change Management panel to display the Manage Changes panel.

c. Select option 1 on the CM - Manage Changes panel. The Specify Data Set / Member Information panel is displayed, as shown in the following figure.

```
ADB2C11 n ------------------ CM - Changes ------------------ Row 1 to 9 of 450

Commands: COMMENT  EXPORT
Line commands:
 U - Update  AN - Analyze  RN - Run  VE - Versions  ST - Statements
 PQ - Prerequisites  IG - Ignores  MA - Masks  S - Show WSL  B - Checkpoint
 AT - Associated Targets  ? - Show all line commands

Sel      ID Owner    Name                         Type     Status   I Comment
         * *         *                            *        *          * *
--- -------- -------- -------------------------> -------- -------- - ---------
        845 J148286  JOEDROPGV                    CHANGE   CANCELED

        844 NNAGAI   CHG0926-01                   CHANGE   DEFINED
             ADD CHECK TO MQT
        843 WBELIS   TEST99                       CHANGE   INITIAL

        842 WBELIS   TESTBEL                      CHANGE   DEFINED
```

Figure 480. CM - Changes panel (ADB2C11)

d. Specify the EXPORT command. Panel ADB2C11 is displayed, as shown in , then continue with the remaining steps.

Results

The changes are exported to the specified data set.

Importing multi-target information from a data set on the central system

You can use a batch interface to process a status update file on a central multi-target system so that the central system will be synchronized with the target systems for the targets that are associated with the central system.

About this task

To import a multi-target change information from a data set:

Procedure

1. Select option 1 on the Change Management panel to display the Manage Changes panel as shown in the following figure:

```
ADB2C1 in -------------------- CM - Manage Changes -------------------- 13:27
Option ===>


   1 - Display changes                              DB2 System: DSNB
   2 - Create a change                              DB2 SQL ID: WBELIS
   3 - Create delta for target
   4 - Import changes
   5 - Export changes
   6 - Export multi-target information into a dataset (on target system)
   7 - Import multi-target information from a dataset (on central system)




Enter display selection criteria.  Settings: LIKE operator;  Criteria not saved
Name . . . . .                              > Created by . .          >
Owner  . . . .          >                     Altered by . .          >
Type . . . . .                                Status . . . .
Created within                                Change ID  . .
Altered within
```

Figure 481. Manage Changes panel (ADB2C1)

2. Select option 7, Import multi-target information from a dataset (on central
   system). The Specify Data Set / Member Name panel (ADBPVERD) is
   displayed as shown in the following example:

```
ADBPVERD  ----------------- Specify Data Set / Member Name -----------------

*Data Set Name . . MTC.UPDATES
*Member Name . . .


 F1=HELP      F2=SPLIT     F3=END     F4=expand   F5=RFIND    F6=RCHANGE
 F7=UP        F8=DOWN      F9=SWAP    F10=LEFT     F11=RIGHT
```

Figure 482. Specify Data Set / Member Name panel (ADBPVERD)

3. Specify the dataset (and member) where the target information is to be
   exported. A new job is created. When the job is submitted, should update the
   target information on central multi-target system. The appropriate entries in the
   ADBCHGAT table will be updated.

   A report is also generated by the job (dd REPORT) The purpose of the report is
   to allow you to determine what action should be taken for each entry in the
   file. You can specify REPORT_LEVEL=All, Current Location, or Updated.

   All   All entries pertaining to all MTC locations are displayed.

   Current Location
              All entries pertaining to the current location are displayed.

   Updated
              Only updated entries are displayed. The entries are a subset of the
              entries for the current location. This is the default setting.

   If a search criteria results in no rows found, a warning is displayed and the
   process ends with RC=8.

Making changes using Change Management batch interface

Change Management batch interface enables you to create, customize, and reuse batch jobs when managing changes in DB2 Admin change management. You can import, analyze, and run changes by submitting batch jobs, and without using the change management ISPF panels.

Topics:
- "Overview: Change Management batch interface"
- "Configuring Change Management batch interface" on page 600
- "How to use the Change Management batch interface" on page 602
- "Using parameters for Change Management batch interface" on page 602
- "Using symbol variables: Change Management batch interface" on page 707
- "Importing changes using the Change Management batch interface" on page 714
- "Using DB2 templates: Change Management batch interface" on page 722
- "Examples: Invoking the Change Management batch interface for various actions" on page 724

Overview: Change Management batch interface

Change Management batch interface is an alternate interface for using the Change Management panels in DB2 Admin to manage changes, and for using the panels in DB2 Object Comparison Tool to run compare to define a change to be managed by DB2 Admin Change Management. Using Change Management batch interface enables you to make changes without using the change management ISPF panels.

The Change Management batch interface can be used to do everything from creating a change to running a change. Using DB2 Admin Change Management terminology, the following functions are supported in the Change Management batch interface:

- Run compare (invokes DB2 Object Comparison Tool to generate a delta change file that can be managed by DB2 Admin Change Management)
- Import mask
- Import ignore
- Import change (equivalent to using register change in the DB2 Admin panels)
- Analyze change (using the automatic method)
- Analyze multi-target change
- Build run job
- Run change
- Recover change

Note: One or more of these functions can be done in one call to the Change Management batch interface, except for the "recover change" function which cannot be done with any other action.

For more information on running compare using the Change Management batch interface, see the "Creating a Change Management batch job to run compare" topic in the *DB2 Object Comparison Guide*.

Change Management batch interface also supports importing one or more DDL or delta change files into a single change.

CAUTION:
If you use Change Management batch interface to import a DDL file, make sure that the first line of the DDL file is a simple SQL comment, meaning that it starts with two dash symbols (--). If the imported DDL file does not begin with a simple SQL comment, import change errors might orccur.

While the Change Management batch interface can be used to manage changes, from creating a change to running a change, it can also be used to manage a change that was created with the DB2 Admin panels. Likewise, a change that was imported using Change Management batch interface can be managed using DB2 Admin panels.

Restriction: The following Change Management functions are not supported using Change Management batch interface:
* Report changes
* Import a version file
* Import a version scope
* Analyze change (using the user-defined or existing base version file method)

Configuring Change Management batch interface

You can optionally configure Change Management batch interface by defining your own JCL symbols as parameters or by customizing the Change Management batch interface JCL procedure name.

Topics:
* "Defining your own JCL symbols as parameters"
* "Customizing the Change Management batch interface JCL procedure name" on page 601

Defining your own JCL symbols as parameters

Some customization of the Change Management batch interface JCL procedure is required if you want to define your own JCL symbols for the Change Management batch interface JCL procedure.

About this task

When you invoke Change Management batch interface, you use a JCL EXEC statement, such as:

```
//GOCCM  EXEC GOCCM,SSID=DSNA,PLAN=ADB
```

Procedure

If you customized the Change Management batch interface JCL procedure so that other JCL symbols are allowed on the EXEC statement, some additional configuration is required. Use the following example for guidance.

Example

The user-defined JCL symbols on the PROC statement are P1 and P2, so the values of P1 and P2 are specified in PARM as follows:

Content of member GOCCM:

```
//GOCCM  PROC SSID=,PLAN=,P1=,P2=
//GOCCM  EXEC PGM=IKJEFT01,DYNAMNBR=200,
// PARM=('CALL *(GOCCCM) ''/SSID(&SSID) PLAN(&PLAN) '
```

```
// 'P1=&P1 P2=&P2',
// '''')
<snip>
//GOCCM  PEND
```

The EXEC JCL statement you specify to invoke Change Management batch interface to analyze and build a run job for a change is:

```
//GOCCM  EXEC GOCCM,SSID=DSNA,PLAN=ADB,P1=ABC,P2=XYZ
```

By updating the PARM to contain the references to P1 and P2, any JCL job that Change Management batch interface creates contains the P1 and P2 JCL symbols on the EXEC statement. The EXEC JCL statement generated by Change Management batch interface in the run job contains the values for P1 and P2, as follows:

```
//GOCCM  EXEC GOCCM,SSID=DSNA,PLAN=ADB,P1=ABC,P2=XYZ
```

Customizing the Change Management batch interface JCL procedure name

You must customize the Change Management batch interface JCL procedure name if you use a member name other than the default, GOCCM.

About this task

The default Change Management batch interface JCL procedure name is GOCCM. A cataloged procedure name is a member or alias of a PDS or PDSE that is defined in your environment to be the libraries that store cataloged JCL procedures.

Procedure

If you use a member name other than GOCCM, you must customize the Change Management batch interface JCL procedure. Use the following example for guidance.

Example

The Change Management batch interface JCL procedure is stored in member TEST01 instead of the default GOCCM. Since the default member name is not used, the **MBR** parameter must be set in PARM, as follows:

Content of member TEST01:
```
//GOCCM  PROC SSID=,PLAN=,P1=,P2=
//GOCCM  EXEC PGM=IKJEFT01,DYNAMNBR=200,
// PARM=('CALL *(GOCCCM) ''/SSID(&SSID) PLAN(&PLAN) ',
// 'MBR=TEST01 P1=&P1 P2=&P2',
// '''')
<snip>
//GOCCM  PEND
```

The EXEC JCL statement you specify to invoke Change Management batch interface using the TEST01 cataloged JCL procedure to analyze and build run job for a change is:

```
//GOCCM  EXEC TEST01,SSID=DSNA,PLAN=ADB,P1=ABC,P2=XYZ
```

The EXEC JCL statement generated by Change Management batch interface in the run job uses the TEST01 cataloged procedure, as follows:

```
//GOCCM  EXEC TEST01,SSID=DSNA,PLAN=ADB,P1=ABC,P2=XYZ
```

How to use the Change Management batch interface

The Change Management batch interface is a JCL procedure. Using a JCL procedure gives you the flexibility to define required DD statements using JCL symbols.

You invoke the JCL procedure to enable Change Management batch interface. You can use the same JCL procedure for single or multiple DB2 subsystems.

See "Examples: Invoking the Change Management batch interface for various actions" on page 724

Using parameters for Change Management batch interface

The Change Management batch interface contains a list of parameters that allows you to control how changes are managed.

Topics:
- "Parameter syntax for Change Management batch interface"
- "How parameters work: Change Management batch interface" on page 603
- "Parameter definitions: Change Management batch interface" on page 604
- "Using parameter profiles: Change Management batch interface" on page 706

Parameter syntax for Change Management batch interface

The following sections describe how the Change Management batch interface parameter syntax works.

Use of quotes

Use of upper-case or lower-case

Defining a user symbol

Specifying a fully qualified data set name

Using DB2 Admin data set template parameters

Use of quotes

The Change Management batch interface parameter syntax must be contained within single quotes, as follows:

```
parameter_name = 'parameter_value'
```

When specifying the fully qualified PDS name, you must enclose the PDS name using double quotes within single quotes. For example, when the WSL PDS is named HLQ.BATCH.WSL, specify the following

When specifying the fully qualified PDS name, enclose the PDS name using double quotes within single quotes. For example, when the WSL PDS is named HLQ.BATCH.WSL specify the following:

```
PDS_FOR_WSL='''HLQ.BATCH.WSL'''
```

Use of upper-case or lower-case

Most parameter values are not case sensitive. However, the following parameter values are case sensitive:

- symbol parameters
- parameters related to data set names
- parameters related to an object owner, name, or comment

Defining a user symbol

When defining a user specified symbol using the symbol_name and symbol_value parameters, a '&' must begin the symbol name and a '.' must end the symbol name. A semi-colon must be specified right after the symbol value, as follows:

```
symbol_name = '&TASK#.' symbol_value = 'ABC';
```

Specifying a fully qualified data set name

When specifying a fully qualified data set name, you can either use two single quotes to represent one single quote, or wrap the parameter value using double quotes.

For example, specifying the following:
```
prefix_for_data_sets = 'WALDO1'
pds_for_wsl = 'WALDO2.WSL'
```

produces a WSL data set name of `WALDO1.WALDO2.WSL`.

To have the WSL data set name be just `WALDO2.WSL`, specify one of the following:
- The **prefix_for_data_sets** parameter determines the data set prefix. For example: `prefix_for_data_sets.pds_for_wsl`.
  ```
  prefix_for_data_sets = 'WALDO2'
  pds_for_wsl = 'WSL'
  ```
- There are three single quotes before and after the value for `pds_for_wsl`.
  ```
  prefix_for_data_sets = 'WALDO1'
  pds_for_wsl = '''WALDO2.WSL'''
  ```

Using DB2 Admin data set template parameters

The parameters that begin with `admin_dataset` can be used to override some of the product default attributes for the types of data sets listed for `admin_dataset_type`. When you use these parameters, the **admin_dataset_type** parameter must be specified with one of the valid values and must be grouped together with one or more of the other `admin_dataset` parameters. The group must be ended with a semi-colon. For more information about the definition of **admin_dataset_type** parameters, see "Parameter definitions: Change Management batch interface" on page 604.

For example:
```
admin_dataset_type = 'CHG'
admin_dataset_dsn = 'CHG.T&TIME.'
admin_dataset_space_priqty = '20';
```

How parameters work: Change Management batch interface
The Change Management batch interface contains a list of parameters that enable you to control various aspects of managing changes, including what action the Change Management batch interface performs when called.

The following sections describe some common Change Management settings and actions you can control with Change Management batch interface parameters. For a full list of Change Management batch interface parameters, see "Parameter definitions: Change Management batch interface."

Using Change Management batch interface

The Change Management batch interface parameters enable you to customize various aspects of managing a change, such as:

- Data set prefixes for data sets dynamically created by the Change Management batch interface
- PDS name to store work statement list (WSL) files
- PDS name to store JCL run jobs for running changes
- Default "change owner" name to use when creating a new change
- Default "change name" to use when creating a new change
- Analyze reporting options
- Utility options
- Admin templates

Batch interface parameters for Change Management actions

You can use the following Change Management batch interface parameters to control what action Change Management batch interface performs.

Table 19. Action parameters for Change Management batch interface

Action	Parameter name	Parameter values
Run compare	action_compare	Y, N
Analyze change	action_analyze_change	Y, N, C
Build run job	action_build_run_job	Y, N
Generate DDL	action_generate_ddl_from_base_version	BEFORE_RUN, AFTER_RUN, SOURCE, TARGET, USER, NO
Generate base version	action_generate_base_version	AUTO, USER, NO
Import change	action_import_change	Y, N
Import ignore	action_import_ignore	Y, N
Import mask	action_import_mask	Y, N
Run change	action_run_change	Y, N
Recover change	action_recover_change	Y, N

For more information on running compare using the Change Management batch interface, see the "Creating a Change Management batch job to run compare" topic in the *DB2 Object Comparison Guide*.

Parameter definitions: Change Management batch interface

The following Change Management batch interface parameters can be used to control Change Management actions and settings.

Change Management batch interface parameters: listed alphabetically

action_analyze_change
> The **action_analyze_change** parameter specifies whether to analyze a change. If a change is also being imported, the change that is analyzed is the newly imported change. Otherwise, the change to be analyzed is identified by the **change_owner** and **change_name** parameters.

Values:

 Y Specifies that the change identified by the **change_owner** and **change_name** parameters is analyzed. If a change is also being imported, the change that is analyzed is the change identified by the **new_change_owner** and **new_change_name** parameters. A change that is already in ANALYZED state is reanalyzed.

 N Specifies that no change is analyzed.

 C Same as the **Y** parameter value except that the analyze is not done if the change is already in the ANALYZED state. For the **C** parameter, return codes of 0 and 1 are defined as follows:

 - 0 indicates that analyze is not done because the change is already in the ANALYZED state.
 - 1 indicates that analyze is done, and no warnings or errors are issued. The state is the same the state of `action_analyze_change = 'Y'` with RC=0.

 There are no changes to the meanings of other RC values.

 blank Specifies that this parameter defaults to Y if a change is imported during this call to the Change Management batch interface.

Default:
 blank

action_build_run_job
 The **action_build_run_job** parameter specifies whether a run job is created for a change. If a change is also being imported, a run job is created for the newly imported change. Otherwise, a run job is created for the change identified by the **change_owner** and **change_name** parameters.

 Values:

 Y Specifies that a run job is created for the change identified by the **change_owner** and **change_name** parameters. If a change is also being analyzed, a run job is created for the change after it is analyzed.

 N Specifies to not create a run job for the change.

 blank Specifies that this parameter should default to Y if a change is analyzed during this call to the Change Management batch interface.

Default:
 blank

action_cancel_change
 The **action_cancel_change** parameter specifies whether to cancel a change specified by the **change_owner** and **change_name** parameters. No other CM Batch actions are allowed when you are requesting a change to be canceled. Any type of change can be canceled except for a multi-target change and a change that is in COMPLETE state.

 Values

 U Specifies an unconditional cancel change. The specified change is canceled even if other changes have the change

as a prerequisite. If other changes have the change as a prerequisite, the changes that depend on the change that is being canceled are set to DEFINED status and must be analyzed before being run. A list of changes that have the change to be cancelled as a prerequisite are listed.

C   Specifies a conditional cancel change. The specified change is canceled if no other changes have the change as a prerequisite change. If other changes have the change as a prerequisite change, an error message is issued. A list of changes that have the change to be cancelled as a prerequisite are listed.

N   Specifies to not cancel the change.

Default
      N

action_compare

The **action_compare** parameter specifies whether to run the DB2 Object Comparison Tool to define a change that can be imported and managed by DB2 Admin Change Management.

Values

Y

Specifies to run DB2 Object Comparison Tool to define a change that can be managed by DB2 Admin Change Management. A compare report and a delta change file is generated that can be imported as a new change. The delta change file attributes are taken from the parameters for **admin_dataset_type** = 'DELTA'.

By specifying **action_compare** = 'Y' and **action_import_change** = 'N', you can run DB2 Object Comparison Tool to just generate a compare report and delta change file, without importing the result as a change. This setting enables you to view the differences between the compare source and target, and perhaps run the compare multiples times to fine-tune the differences between the source and target. When no more compares are needed and the change is ready to be deployed, the delta change file can be imported as a new change.

Note: The files with DD names that start with IMCHG are not used.

N   Specifies to not run DB2 Object Comparison Tool to define the change.

Default
      N

For more information about using the Change Management batch interface to run compare, see the "Creating a Change Management batch job to run compare" topic in the *DB2 Object Comparison Guide*.

action_delete_change

The **action_delete_change** parameter specifies whether to delete a change specified by the **change_owner** and **change_name** parameters. No other CM

Batch actions are allowed when you are requesting a change to be deleted. Any type of change can be deleted except for a multi-target change.

Values

C   Specifies a conditional delete change. The specified change is deleted if no other changes have the change as a prerequisite. If other changes have the change as a prerequisite, an error message is issued. A list of changes that have the change to be deleted as a prerequisite are listed.

N   Specifies to not delete the change.

U   Specifies an unconditional delete change. The specified change is deleted even if other changes have the change as a prerequisite. If other changes have the change as a prerequisite, the changes that depend on the change that is being deleted are set to DEFINED status and must be analyzed before being run. A list of changes that have the change to be deleted as a prerequisite are listed.

Default

N

action_delete_mask

The **action_delete_mask** parameter specifies whether to delete the mask specified by the **mask_owner** and **mask_name** parameters. No other CM Batch actions are allowed when you are requesting a mask to be deleted.

Values

C   Delete the mask if it is not associated with a registered change that needs the mask for implementation. If the mask is needed by one or more changes for implementation, the names of changes are displayed and the mask is not deleted.

CONDITIONAL mode does not cover the scenario in which changes can have a mask associated with it but the mask is no longer needed to implement the change. For example, when a change is imported using masking, after the import completes the mask is no longer needed to implement that change.

N   The delete mask action is not enabled.

U   Delete the mask even if it is associated with a registered change that needs the mask for implementation. The names of changes that need the mask for implementation are displayed and the mask is deleted.

UNCONDITIONAL mode does not cover the scenario in which changes can have a mask associated with it but the mask is no longer needed to implement the change. For example, when a change is imported using masking, after the import completes the mask is no longer needed to implement that change. In this scenario, UNCONDITIONAL does not report the imported change.

Default

N

action_generate_base_version

The **action_generate_base_version** parameter specifies whether and how to generate a base version. This parameter enables you to start Change Management batch interface only to generate a base version. The **generate_base_version_before_run** and **generate_base_version_after_run** parameters enable you to configure Change Management batch interface so that base versions are automatically generated during the run change process. Generating a base version by using the **action_generate_base_version** parameter and choosing to generate DDL from a base version (**generate_ddl_from_base_version** parameter) in the same invocation of Change Management batch interface enables to you save the current definitions of objects in the base version and also to generate a DDL file from these object definitions.

Values:

AUTO
A base version is generated and the content is automatically determined by the product for the specified change entry. The content of the base version is based on the registered change statements for the specified change entry.

USER A base version is generated and the content is determined by a user-specified version scope.

NO A base version is not generated. However, this setting has no control over whether a base version is automatically generated as determined by the **generate_base_version_before_run** and **generate_base_version_after_run** parameters.

Default:
NO

action_generate_ddl_from_base_version

The **action_generate_ddl_from_base_version** parameter specifies whether to generate DDL and from a base version. The generated file must be run by using the DB2 Admin ADBTEP2 program. See ADBTEPR SAMP member for a sample job of running ADBTEP2.

Values:

BEFORE_RUN
DDL and DB2 Admin statements are generated for the base version that was created before the specified change was implemented.

AFTER_RUN
DDL and DB2 Admin statements are generated for the base version that was created after the specified the change was implemented.

SOURCE
DDL and DB2 Admin statements are generated for the base version that is recorded as the source base version for the specified change.

TARGET
DDL and DB2 Admin statements are generated for the base version that is recorded as the target base version for the specified change.

USER DDL and DB2 Admin statements are generated for the user-specified base version that is identified by the `base_version_owner` and `base_version_name` parameters.

NO DDL and DB2 Admin statements are not generated for any base version.

Default
NO

action_import_change
The **action_import_change** parameter specifies whether a change is imported. If **action_compare = 'N'**, the DDL or delta change files that are defined by the files that begin with IMCHG (for example, IMCHG001, IMCHG002, and so on) are imported as a new change. If **action_compare = 'Y'**, the result of the compare is imported as a new change.

Values:

Y If **action_compare = 'N'**, specifies that the content of files IMCHG001 up through IMCHG999 are imported into a new change. You do not need to define all of the IMCHG* files must be defined. For example, only 2 DDL files or delta change files is imported, you need to define only IMCHG001 and IMCHG002.

If action_compare = 'Y', specifies that the result of the compare is imported as a new change. The contents of the files with names IMCHG001 through IMCHG999 are not imported as a new change.

N Specifies that no importing of a change is done.

blank Specifies that this parameter defaults to Y if either of the following is true:
1. **action_compare = 'N'**, and the IMCHG001 DD is defined and not empty.
2. **action_compare = 'Y'**.

Default:
blank

action_import_ignore
The **action_import_ignore** parameter specifies whether an ignore that is defined by the IMIGNORE DD statement is imported as a new ignore.

Values:

Y Specifies that the content of the IMIGNORE DD statement is imported into a new ignore.

N Specifies that no importing of an ignore is done.

blank Specifies that this parameter defaults to Y if the IMIGNORE DD statement is defined and not empty.

Default:
blank

action_import_mask

The **action_import_mask** parameter specifies whether a mask that is defined by the IMMASK DD is imported as a new mask.

Values:

    Y   Specifies that the content of the IMMASK DD statement is imported into a new mask.

    N   Specifies that no importing of a mask is done.

    blank Specifies that this parameter defaults to Y if the IMMASK DD statement is defined and not empty.

Default:

    blank

action_recover_change

The **action_recover_change** parameter specifies whether to recover the change.

Values:

    Y   Specifies to recover the change.

    N   Specifies to not recover the change.

Default:

    N

action_run_change

The **action_run_change** parameter specifies whether to run the change. If a change is also being imported, the change that is run is the newly imported change. Otherwise, the change to be run is identified by the **change_owner** and **change_name** parameters.

Values:

    Y   Specifies to run the change.

    N   Specifies to not run the change.

Default:

    N

adbtep2_ac

The **adbtep2_ac** parameter specifies whether to use autocheck when a change is run. Certain SQL or utility operations can place an object into check-pending state. If you set the Autocheck (AC) parameter value to YES, run change (ADBTEP2) tracks the statements and processes that can place an object in check-pending. If one of these statements is encountered while running a change, an automatic CHECK DATA is done to remove the check-pending state. For the complete description see Chapter 16, "Using the Batch Restart programs: ADBTEP2 and ADBTEPA," on page 387.

Values:

    YES   The automatic check process is performed.

    NO   The automatic check process is not performed.

Default:

    NO

adbtep2_advisoryautorebuild

The **adbtep2_advisoryautorebuild** parameter specifies whether the product, when a change is run, initiates a REBUILD when an object is in certain rebuild pending states. For the complete description and list of values see Chapter 16, "Using the Batch Restart programs: ADBTEP2 and ADBTEPA," on page 387.

Values:

YES   The product automatically attempts a REBUILD if the object is in the ARBDP state.

However, if the parameter **run_reorg_rebuild** was specified as 'A - All relevant' to generate an explicit REBUILD during the change flow, then the value NO is passed to ADBTEP2. The NO value prevents an automatic REBUILD that duplicates the explicit REBUILD.

NO   The product does not automatically attempt a REBUILD if the object is in the ARBDP state.

Default:
NO

adbtep2_advisoryautoreorg

The **adbtep2_advisoryautoreorg** parameter specifies whether the product, when a change is run, initiates a REORG when an object is in certain reorganization-pending states. For the complete description see Chapter 16, "Using the Batch Restart programs: ADBTEP2 and ADBTEPA," on page 387.

Values:

YES   The product automatically attempts a REORG if the object is in AREOR or AREO* state.

However, if the parameter **run_reorg_rebuild** was specified as 'A - All relevant' to generate an explicit REORG during the change flow, then the value NO is passed to ADBTEP2. The NO value prevents an automatic REORG that duplicates the explicit REORG.

NO   The product does not automatically attempt a REORG if the object is in AREOR or AREO* state.

Default:
NO

adbtep2_autorebuild

The **adbtep2_autorebuild** parameter specifies whether the product, when a change is run, initiates a REBUILD when an object is in certain rebuild pending states. For the complete description see Chapter 16, "Using the Batch Restart programs: ADBTEP2 and ADBTEPA," on page 387.

Values:

YES   The product automatically attempts a REBUILD if the object is in one of these states: RBDP, RBDP*, or PSRBD state.

However, if the parameter **run_reorg_rebuild** was specified as 'M - Mandatory' or 'A - All relevant' to generate an explicit REBUILD during the change flow, then

the value NO is passed to ADBTEP2. The NO value
prevents an automatic REBUILD that duplicates the explicit
REBUILD.

NO The product does not automatically attempt a REBUILD if
the object is in one of these states: RBDP, RBDP*, or PSRBD
state.

Default:
YES

adbtep2_autoreorg

The **adbtep2_autoreorg** parameter specifies whether the product, when a
change is run, initiates a REORG when an object is in certain
reorganization-pending states. For the complete description see Chapter 16,
"Using the Batch Restart programs: ADBTEP2 and ADBTEPA," on page
387.

Values:

YES The product automatically attempts a REORG if the object
is in the REORP state.

However, if the parameter **run_reorg_rebuild** was
specified as 'M - Mandatory' or 'A - All relevant' to
generate an explicit REORG during the change flow, then
the value NO is passed to ADBTEP2. The NO value
prevents an automatic REORG that duplicates the explicit
REORG.

NO The product does not automatically attempt a REORG if
the object is in the REORP state.

Default:
YES

adbtep2_binderror

The **adbtep2_binderror** parameter specifies how BIND or REBIND errors
are handled when running a change. For the complete description see
Chapter 16, "Using the Batch Restart programs: ADBTEP2 and ADBTEPA,"
on page 387.

Values:

MAXE
The failed command is written to the ADBHOLD table.
The MAXERROR setting determines if the processing stops
immediately, after *nn* errors, or if the bind error does not
stop processing.

SAVE The failed command is written to the ADBHOLD table.
Processing continues.

IGNORE
The failed command is not written to the ADBHOLD table.
Processing continues.

Default:
MAXE

adbtep2_pendingchangescheck

The **adbtep2_pendingchangescheck** parameter specifies whether a check is
made when a change is run to avoid losing any DB2 pending changes as

part of a DROP action. This function is supported on DB2 V10 or later. For the complete description see Chapter 16, "Using the Batch Restart programs: ADBTEP2 and ADBTEPA," on page 387.

Values:

YES The DROP is not performed if a DB2 pending change exists.

NO The DROP is performed without checking for pending changes.

Default:
NO

adbtep2_restart

The **adbtep2_restart** parameter specifies the RESTART value that is passed to the ADBTEP2 (adbtepx) program. You can restart a change at the beginning of the change work list or at the point where the change stopped running in a previous run.

Values:

Y RESTART(YES) is used when ADBTEP2 is called.

N RESTART(NO) is used when ADBTEP2 is called.

Default:
Y

adbtep2_stogroup_auto_reorg_rebuild

The **adbtep2_stogroup_auto_reorg_rebuild** parameter specifies whether the product, when a change is run, initiates a REORG or REBUILD for the table space or index to implement the effect of altering STOGROUP attribute. For the complete description and list of values see Using the Batch Restart programs: ADBTEP2 and ADBTEPA.

Values:

YES The product automatically attempts a REORG or REBUILD for the table space or index after SQL statement ALTER STOGROUP is executed. However, if the parameter run_reorg_rebuild was specified as 'A - All relevant' to generate an explicit REORG or REBUILD during the change flow, then the value NO is passed to ADBTEP2. The NO value prevents an automatic REORG or REBUILD that duplicates the explicit REORG or REBUILD.

NO The product does not automatically attempt a REORG or REBUILD for the table space or index after SQL statement ALTER STOGROUP is executed.

Default:
NO

admin_dataset_bufno

The **admin_dataset_bufno** parameter specifies the BUFNO attribute of the TSO ALLOCATE statement for the DB2 Admin data set. The BUFNO attribute is for the number of buffers to be assigned for data control blocks. For more information, see the admin_dataset_type parameter.

Values:
An integer value 1-255, blank

> **blank** The BUFNO attribute is not specified for the ALLOCATE statement.

> **Default:**
> blank

admin_dataset_dataclas
> The **admin_dataset_dataclas** parameter specifies the DATACLAS attribute of the TSO ALLOCATE statement for the DB2 Admin data set. The DATACLAS attribute is for the data class name. For more information, see the admin_dataset_type parameter.

> **Values:**

>> **A valid data class name**
>> DB2 Admin does not validate this value. If an invalid value is specified, an error message is generated from TSO when the allocate of the data set is attempted. DB2 Admin then sets the RECFM, LRECL, and BLKSIZE attributes by specifying these attributes on the ALLOCATE statement. By default, DB2 Admin specifies the space attributes on the allocate statement but you can omit the space attributes from the ALLOCATE statement by specifying admin_dataset_space_priqty = '<NONE>' for the DB2 Admin.

>> **blank** The DATACLAS attribute is not specified for the ALLOCATE statement.

> **Default:**
> blank

admin_dataset_device_unit
> The **admin_dataset_device_unit** parameter specifies the device unit for the DB2 Admin data set. For more information, see the admin_dataset_type parameter.

> **Values:**
> A valid device unit, <NONE>

>> **<NONE>**
>> Specifies that the UNIT clause is omitted from the ALLOCATE statement.

> **Default:**
> space_unit_name

admin_dataset_dir
> The **admin_dataset_dir** parameter specifies the DIR attribute of the TSO ALLOCATE statement for the DB2 Admin data set. The DIR attribute is for the number of directory blocks. For more information, see the admin_dataset_type parameter. This parameter is only used for the following types of DB2 Admin data sets: IFF, DELTA, DDL SRCVF, TGTVF, MTC.

> If the SPACE(priqty,secqty) clause is omitted, then no default value is specified.

> **Values:**
> An integer greater than zero, blank

>> **blank** If the SPACE(priqty,secqty) clause is not to be omitted,

specifies that the following default values are used for the DB2 Admin data set type that is in effect:

- IFF: 60. A user specified value for directory blocks that you specify only if the DB2 Admin default is insufficient for the change that is being analyzed.
- DELTA: 60
- DDL: 60
- SRCVF: 60
- TGTVF: 60
- MTC: 60

Default:
blank

admin_dataset_dsn

The **admin_dataset_dsn** parameter specifies the data set name for the DB2 Admin data set. For more information, see the admin_dataset_type parameter.

Values:

A valid data set name.
The data set name can be 1 to 46 characters or blank.

blank   Specifies that the following default values are to be used for the indicated DB2 Admin data set type that is in effect:

- CHG: &SSID..&CHGTAG..CHG
- DDL: &SSID..&CHGTAG..T&TIME..DDL
- DELTA: D&DATE..T&TIME..DELTA
- IFF: &SSID..&CHGTAG..IFF
- MTC: &SSID..D&DATE..T&TIME..MTC
- SRCVF: OC.D&DATE..T&TIME..SRCVF
- TGTVF: OC.D&DATE..T&TIME..TGTVF

Default:
blank

admin_dataset_dsntype

The **admin_dataset_dsntype** parameter specifies the DSNTYPE attribute of the TSO ALLOCATE statement for the DB2 Admin data set. The DSNTYPE attribute is for the type of data set. For more information, see the admin_dataset_type parameter. This parameter is only used for the following types of DB2 Admin data sets: IFF, DELTA, DDL SRCVF, TGTVF, MTC.

Values:
LIBRARY, PDS, blank

blank   For data set type IFF, the default is PDS. Otherwise the DSNTYPE attribute is not added to the ALLOCATE statement.

Default:
blank

admin_dataset_expdt

The **admin_dataset_expdt** parameter specifies the EXPDT attribute of the TSO ALLOCATE statement for the DB2 Admin data set. The EXPDT

attribute is an expiration date. For more information, see the admin_dataset_type parameter. This parameter is mutually exclusive with the **admin_dataset_retpd** parameter.

Values:
> A valid expiration date as defined for the EXPDT attribute for the TSO ALLOCATE statement, blank
>
> **blank** The EXPDT attribute is not specified for the ALLOCATE statement.

Default:
> blank

admin_dataset_maxvol
> The **admin_dataset_maxvol** parameter specifies the MAXVOL attribute of the TSO ALLOCATE statement for the DB2 Admin data set. For more information, see the admin_dataset_type parameter.

Values:

> **A valid maxvol value as defined by the TSO ALLOCATE statement**
> > DB2 Admin does not validate this value. If an invalid value is specified, an error message is generated from TSO when the allocate of the data set is attempted.
>
> **blank** The MAXVOL attribute is not specified for the ALLOCATE statement.

Default:
> blank

admin_dataset_mgmtclas
> The **admin_dataset_mgmtclas** parameter specifies the MGMTCLAS attribute of the TSO ALLOCATE statement for the DB2 Admin data set. The MGMTCLAS attribute is for the management class name. For more information, see the admin_dataset_type parameter.

Values:

> **A valid management class name**
> > DB2 Admin does not validate this value. If an invalid value is specified, an error message is generated from TSO when the allocate of the data set is attempted.
>
> **blank** The MGMTCLAS attribute is not specified for the ALLOCATE statement.

Default:
> blank

admin_dataset_retpd
> The **admin_dataset_retpd** parameter specifies the RETPD attribute of the TSO ALLOCATE statement for the DB2 Admin data set. The RETPD attribute is a retention period specified in number of days. For more information, see the admin_dataset_type parameter. This parameter is mutually exclusive with the **admin_dataset_expdt** parameter.

Values:
> An integer value representing the number of days, blank
>
> **blank** The RETPD attribute is not specified for the ALLOCATE statement.

Default:
blank

admin_dataset_space_priqty

The **admin_dataset_space_priqty** parameter specifies the primary quantity for the DB2 Admin data set. For more information, see the admin_dataset_type parameter.

Values:
A valid PRIQTY value, <NONE>, blank

<NONE>
Specifies that the SPACE(priqty,secqty), unit of space clauses, and space directory attributes be omitted from the ALLOCATE statement.

blank

Specifies that the following default values are to be used for the indicated DB2 Admin data set type that is in effect:
- CHG: 10
- DDL: 10
- DELTA: 10
- IFF: 2
- MTC: 10
- SRCVF: 10
- TGTVF: 10

Default:
blank

admin_dataset_space_secqty

The **admin_dataset_space_secqty** parameter specifies the secondary quantity for the DB2 Admin data set. For more information, see the admin_dataset_type parameter.

If the SPACE(priqty,secqty) clause is omitted, then no default value is specified.

Values:
A valid SECQTY value, blank

blank   If the SPACE(priqty,secqty) clause is not to be omitted, the following default values are used for the DB2 Admin data set type that is in effect:
- CHG: 10
- DDL: 10
- DELTA: 10
- IFF: 2
- MTC: 10
- SRCVF: 10
- TGTVF: 10

Default:
blank

admin_dataset_space_type

The **admin_dataset_space_type** parameter specifies the space unit type for the DB2 Admin data set. For more information, see the admin_dataset_type parameter.

If the SPACE(priqty,secqty) clause is omitted, then no default value is specified.

Values:

CYL Specifies that the space unit type is cylinders.

TRK Specifies that the space unit type is tracks.

blank Specifies that the following default values are used for the DB2 Admin data set type that is in effect:
- CHG: CYL
- DDL: CYL
- DELTA: CYL
- IFF: CYL
- MTC: CYL
- SRCVF: CYL
- TGTVF: CYL

Default:
blank

admin_dataset_storclas

The **admin_dataset_storclas** parameter specifies the STORCLAS attribute of the TSO ALLOCATE statement for the DB2 Admin data set. The STORCLAS attribute is for the storage class name. For more information, see the admin_dataset_type parameter.

Values:

A valid storage class name
DB2 Admin does not validate this value. If an invalid value is specified, an error message is generated from TSO when the allocate of the data set is attempted.

blank The STORCLAS attribute is not specified for the ALLOCATE statement.

Default:
blank

admin_dataset_type

The **admin_dataset_type** parameter specifies the type of data set the other DB2 Admin data set template parameters are for. You can specify multiple DB2 Admin data set types. Separate each type with a semicolon.

Note: You can use the Admin data set templates to override the default values for some data sets that are used to process a change. The data set types supported with these parameters are: CHG, DDL, DELTA, IFF, MTC, SRCVF, and TGTVF.

The following parameters are DB2 Admin data set template parameters:
- **admin_dataset_bufno**
- **admin_dataset_dataclas**
- **admin_dataset_device_unit**

- **admin_dataset_dir**
- **admin_dataset_dsn**
- **admin_dataset_dsntype**
- **admin_dataset_expdt**
- **aadmin_dataset_maxvol**
- **admin_dataset_mgmtclas**
- **admin_dataset_retpd**
- **admin_dataset_space_priqty**
- **admin_dataset_space_secqty**
- **admin_dataset_space_type**
- **admin_dataset_storclas**
- **admin_dataset_type**
- **admin_dataset_volume**

Values:

CHG   Specifies that the DB2 Admin data set template parameters that are specified before the next semicolon in the parameter list are for the CHG DB2 Admin data set.

DDL   Specifies that the Admin data set template parameters that are specified before the next semicolon in the parameter list are for the DDL Admin data set. This data set is the output data set when generating DDL from a base version.

DELTA
      Specifies that the Admin data set template parameters that are specified before the next semicolon in the parameter list is for the compare delta change file.

IFF   Specifies that the DB2 Admin data set template parameters that are specified before the next semicolon in the parameter list are for the IFF PDS DB2 Admin data set.

MTC   Specifies that the Admin data set template parameters that are specified before the next semicolon in the parameter list are for the multi-target change file.

SRCVF
      Specifies that the Admin data set template parameters that are specified before the next semicolon in the parameter list are for a compare source version work file.

TGTVF
      Specifies that the Admin data set template parameters that are specified before the next semicolon in the parameter list are for a compare target version work file.

blank   Specifies that the DB2 Admin data set template parameters are ignored until a supported value for **admin_dataset_type** is specified.

Default:
      blank

admin_dataset_volume
      The **admin_dataset_volume** parameter the VOLUME attribute of the TSO

ALLOCATE statement for the DB2 Admin data set. For more information, see the admin_dataset_type parameter.

Values:
> One or more serial numbers separated by a comma, blank
>
> **blank** The VOLUME attribute is not specified for the ALLOCATE statement.

Default:
> blank

allow_implicit_drop_of_excluded_objects
> The `allow_implicit_drop_of_excluded_objects` parameter specifies whether excluded objects can be dropped implicitly.
>
> **Values:**
>> **YES** Excluded objects can be dropped implicitly.
>>
>> **NO** Excluded objects cannot be dropped implicitly.
>
> **Default:**
>> NO

allow_rotate_parts
> The `allow_rotate_parts` parameter specifies whether to generate the rotate partition or alter partition statement when the condition for a rotate is met.
>
> **Values:**
>> **Y** Generate the rotate partition statement. Data from the rotating partitions is unloaded before the rotate takes place. You can either reload the data or discard it.
>>
>> **N** Generate the alter partition statement. Data from the rotating partitions is reloaded into the table. Logical and physical partitions are preserved.
>
> **Default:**
>> Y

auth_switch_secadm
> The `auth_switch_secadm` parameter specifies the SECADM authority to use when auth-switching is enabled. The SECADM authority is used to manage all security-related tasks. This parameter applies only if the facility has been enabled for the subsystem as part of the customization process, and applies only when DB2 Admin is connected to DB2 V10 or later.
>
> **Values:**
>> **An SQLID with SECADM authority**
>>> Specify a SECADM authority to manage all security-related tasks.
>
> **Default:**
>> blank

auth_switch_userid
> The `auth_switch_userid` parameter specifies the auth-switch ID to use when auth-switching is enabled. This parameter applies only when the facility has been enabled for the subsystem as part of customization process.
>
> **Values:**

An SQLID

> The ID to connect as when auth-switching.

<NONE>

> Avoids producing auth-switch work-statement lists (WSL).

<SQLID>

> Enables the SQLID authorization switching feature.

blank Produces auth-switch WSL, with the ID portion of the WSL as comments.

Default:

> <NONE>

auth_switching_enabled

> The **auth_switching_enabled** parameter specifies whether auth-switching is enabled.

Values:

> **Y**   Auth-switching is used if an auth-switch ID is specified.
>
> **N**   Auth-switching is used.

Default:

> N

base_version_name

> The **base_version_name** parameter specifies the name of the base version to perform the action on. If a base version is being saved or generated, this parameter specifies the name for the new base version if the other base version name parameters are blank. If a base version is not being saved or generated, the value of this parameter must identify the name of an existing base version.

> The base version parameter hierarchy is as follows:
> - base_version_owner
>   - new_base_version_owner
>   - base_version_owner_before_run
>   - base_version_owner_after_run
> - base_version_name
>   - new_base_version_name
>   - base_version_name_before_run
>   - base_version_name_after_run

Values:

> A valid 1- to 128-character version name.

Default:

> AUTO:&CURTS.

base_version_name_after_run

> The **base_version_name_after_run** parameter specifies the name for a new base version that is created after a change is implemented.

> The base version parameter hierarchy is as follows:
> - base_version_owner
>   - new_base_version_owner
>   - base_version_owner_before_run

- base_version_owner_after_run
- base_version_name
  - new_base_version_name
    - base_version_name_before_run
    - base_version_name_after_run

Values:
>Valid version name; 1 to 128 characters

Default:
>new_base_version_name

base_version_name_before_run
>The **base_version_name_after_run** parameter specifies the name for a new base version that is created before a change is implemented.
>
>Base version parameter hierarchy:
>- base_version_owner
> - new_base_version_owner
>   - base_version_owner_before_run
>   - base_version_owner_after_run
>- base_version_name
> - new_base_version_name
>   - base_version_name_before_run
>   - base_version_name_after_run
>
>**Values:**
>>Valid version name; 1 to 128 characters
>
>**Default:**
>>new_base_version_name

base_version_owner
>The base_version_owner parameter specifies the owner of the base version to perform the action on. If a base version is being saved or generated, this parameter specifies the owner for the new base version if the other base version owner parameters are blank. If a base version is not being saved or generated, the value of this parameter must identify the owner of an existing base version.
>
>Base version parameter hierarchy:
>- base_version_owner
> - new_base_version_owner
>   - base_version_owner_before_run
>   - base_version_owner_after_run
>- base_version_name
> - new_base_version_name
>   - base_version_name_before_run
>   - base_version_name_after_run
>
>**Values:**
>>Valid version name; 1 to 128 characters
>
>**Default:**
>>&CURSQLID.

base_version_owner_after
　　　　The base_version_owner_after parameter specifies the owner for a new base version that is created after a change is implemented.

　　　　Base version parameter hierarchy:
- base_version_owner
  - new_base_version_owner
    - base_version_owner_before_run
    - base_version_owner_after_run
- base_version_name
  - new_base_version_name
    - base_version_name_before_run
    - base_version_name_after_run

　　　　Values:
　　　　　　　　Valid version name; 1 to 128 characters

　　　　Default:
　　　　　　　　new_base_version_owner

base_version_owner_before_run
　　　　The base_version_owner_before_run parameter specifies the owner for a new base version that is created before a change is implemented.

　　　　Base version parameter hierarchy:
- base_version_owner
  - new_base_version_owner
    - base_version_owner_before_run
    - base_version_owner_after_run
- base_version_name
  - new_base_version_name
    - base_version_name_before_run
    - base_version_name_after_run

　　　　Values:
　　　　　　　　Valid version name; 1 to 128 characters

　　　　Default:
　　　　　　　　new_base_version_owner

base_version_scope_name
　　　　The base_version_scope_name parameter specifies the default name of an existing version scope to use when generating a new base version using the USER method.

　　　　Base version scope parameter hierarchy:
- base_version_scope_owner
  - base_version_scope_owner_before_run
  - base_version_scope_owner_after_run
- base_version_scope_name
  - base_version_scope_name_before_run
  - base_version_scope_name_after_run

　　　　Values:
　　　　　　　　Valid version name; 1 to 128 characters, blank

Default:
> blank

base_version_scope_name_after_run
> The base_version_scope_name_after_run parameter specifies the name of an existing version scope to use when generating a new base version after a change is implemented. This applies only if the new base version is created using the USER method.
>
> Base version scope parameter hierarchy:
> * base_version_scope_owner
>   - base_version_scope_owner_before_run
>   - base_version_scope_owner_after_run
> * base_version_scope_name
>   - base_version_scope_name_before_run
>   - base_version_scope_name_after_run
>
> **Values:**
>> Valid version name; 1 to 128 characters
>
> **Default:**
>> base_version_scope_name

base_version_scope_name_before_run
> The base_version_scope_name_before_run parameter specifies the name of an existing version scope to use when generating a new base version before a change is implemented. This applies only if the new base version is created using the USER method.
>
> Base version scope parameter hierarchy:
> * base_version_scope_owner
>   - base_version_scope_owner_before_run
>   - base_version_scope_owner_after_run
> * base_version_scope_name
>   - base_version_scope_name_before_run
>   - base_version_scope_name_after_run
>
> **Values:**
>> Valid version name; 1 to 128 characters
>
> **Default:**
>> base_version_scope_name

base_version_scope_owner
> The base_version_scope_owner parameter specifies the default owner of an existing version scope to use when generating a new base version using the USER method.
>
> Base version scope parameter hierarchy:
> * base_version_scope_owner
>   - base_version_scope_owner_before_run
>   - base_version_scope_owner_after_run
> * base_version_scope_name
>   - base_version_scope_name_before_run
>   - base_version_scope_name_after_run

Values:
> Valid version scope owner; 1 to 128 characters

Default:
> &CURSQLID.

base_version_scope_owner_after_run
> The base_version_scope_owner_after_run parameter specifies the owner of an existing version scope to use when generating a new base version after a change is implemented. This applies only if the new base version is created using the USER method

> Base version scope parameter hierarchy:
> - base_version_scope_owner
>   - base_version_scope_owner_before_run
>   - base_version_scope_owner_after_run
> - base_version_scope_name
>   - base_version_scope_name_before_run
>   - base_version_scope_name_after_run

> **Values:**
> > Valid version scope owner; 1 to 128 characters

> **Default:**
> > base_version_scope_owner

base_version_scope_owner_before_run
> The base_version_scope_owner_before_run parameter specifies the owner of an existing version scope to use when generating a new base version before a change is implemented. This applies only if the new base version is created using the USER method

> Base version scope parameter hierarchy:
> - base_version_scope_owner
>   - base_version_scope_owner_before_run
>   - base_version_scope_owner_after_run
> - base_version_scope_name
>   - base_version_scope_name_before_run
>   - base_version_scope_name_after_run

> **Values:**
> > Valid version scope owner; 1 to 128 characters

> **Default:**
> > base_version_scope_owner

change_comment
> The **change_comment** parameter specifies the comment for a new change.

> **Values:**
> > 1 to 128 characters

> **Default:**
> > blank

change_name
> The **change_name** parameter specifies the name of the change to perform the action on. If a change is being imported, this parameter specifies the name for the new change if the value for the **new_change_name** parameter is

blank. If a change is not being imported, the value for this parameter must identify the name of an existing change.

Values:
> Valid change name; 1 to 128 characters

Default:
> AUTO:&CURTS.

change_owner
> The **change_owner** parameter specifies the owner of the change to perform the action on. If a change is being imported, this parameter specifies the owner for the new change if the value for the **new_change_owner** parameter is blank. If a change is not being imported, the value for this parameter must identify the owner of an existing change.

Values:
> Valid change owner; 1 to 128 characters

Default:
> &CURSQLID.

chgtag_type
> The **chgtag_type** parameter specifies the type of values that the DB2 Admin &CHGTAG. symbol variable resolves to. Refer to the product-defined symbol variables information and the definition of the DB2 Admin &CHGTAG. symbol variable for details.

Values:

> **ID** The &CHGTAG. symbol variable value is based on the DB2 Admin generated change ID.

> **NAME**
> > The &CHGTAG. symbol variable value is based on the user specified change name.

> **OWNER**
> > The &CHGTAG. symbol variable value is based on the user specified change owner.

Default:
> ID

compare_ignore_changes_name
> The **compare_ignore_changes_name** parameter specifies the name of an existing Ignore Changes Specification that is stored in the Change Management database. The **compare_ignore_changes_owner** and **compare_ignore_changes_name** parameters uniquely identify an Ignore Changes Specification to be used during the compare process.

Values:
> A valid ignore changes name, blank

> **A valid ignore changes name; 1 to 128 characters**
> > Specify a 1- to 128-character Ignore Changes Specification is used during the compare process.

> **blank** Ignore changes is not used during the compare process.

Default:
> blank

compare_ignore_changes_owner

The **compare_ignore_changes_owner** parameter specifies the owner of an existing Ignore Changes Specification that is stored in the Change Management database. The **compare_ignore_changes_owner** and **compare_ignore_changes_name** parameters uniquely identify an Ignore Changes Specification to be used during the compare process.

Values:

Specify a valid 1- to 128-character ignore changes owner.

Default:

&CURSQLID.

compare_ignore_fields_dsn

The **compare_ignore_fields_dsn** parameter specifies the name of a data set that contains the ignore fields to be used during the compare. The **prefix_for_data_sets** parameter is used to qualify the data set name if the specified data set name is not fully qualified. If the compare ignore fields file IGNORES DD is pre-allocated and this parameter is specified, the ignore fields specified by this parameter are used instead of the pre-allocated compare ignore fields file. The **compare_ignore_fields_dsn** and **compare_ignore_fields_name** parameters are mutually exclusive.

Values:

A valid data set name

The data set must contain ignore fields and be a fixed block sequential data set or a member of a partitioned data set with a record length of 80 (RECFM=F*x*, LRECL=80). The input must be in columns 1-72 of the data set.

Syntax:

```
objecttype: field1,field2, .... ,fieldn
```

where **objecttype** is the DB2 catalog table name and fieldn : is the DB2 catalog column to be ignored

Examples:

```
SYSDATABASE: BPOOL
SYSDATABASE: INDEXBP,STGROUP
SYSTABLESPACE: BPOOL
SYSTABLEPART: PQTY,SQTY,STORNAME,VCATNAME
SYSINDEXES: INDEXSPACE
SYSINDEXPART: PQTY,SQTY,STORNAME,VCATNAME
```

Ignore fields are applied to both the target and the source objects before the definitions are compared.
For more information about specifying ignore fields, see the information about translation masks and ignore fields in the *DB2 Object Comparison Tool User's Guide*.

Default:

blank

compare_ignore_fields_name

The **compare_ignore_fields_name** parameter specifies the name of an existing Ignore Fields Specification that is stored in the Change Management database. The **compare_ignore_fields_owner** and **compare_ignore_fields_name** parameters uniquely identify the Ignore Fields Specification to be used during the compare process. If the compare

ignore fields file IGNORES DD is pre-allocated and this parameter is specified, the ignore fields that are specified by this parameter are used instead of the pre-allocated compare ignore fields file. The **compare_ignore_fields_dsn** and **compare_ignore_fields_name** parameters are mutually exclusive.

Values:
>A valid ignore fields name, blank

>**A valid ignore fields name**
>>Specify a 1- to 128-character Ignore Fields name. The specified Ignore Fields Specification is used during the compare process.

Default:
>blank

compare_ignore_fields_owner
>The **compare_ignore_fields_owner** parameter specifies the owner of an existing Ignore Fields Specification that is stored in the Change Management database. The **compare_ignore_fields_owner** and **compare_ignore_fields_name** parameters uniquely identify the Ignore Fields Specification to be used during the compare process.

>**Values:**
>>Specify a 1- to 128-character Ignore Fields owner.

>**Default:**
>>&CURSQLID.

compare_mask_dsn
>The **compare_mask_dsn** parameter specifies the name of a data set that contains the masks to be used for the compare. The **prefix_for_data_sets** parameter is used to qualify the data set name if the specified data set name is not fully qualified. If the compare masks file MASKS DD is pre-allocated and this parameter is specified, the masks that are specified by this parameter are used instead of the pre-allocated compare masks file. The **compare_mask_dsn** and **compare_mask_name** parameters are mutually exclusive.

>**Values:**

>>**A valid data set name**

>>>The data set must contain masks and must be a fixed block sequential data set or a member of a partitioned data set with a record length of 80 (RECFM=Fx, LRECL=80). The input must be in columns 1-72 of the data set.

>>>Here are some mask definition examples:
```
NAME: ABC*, DEF*
NAME: HLQ*D*, NEW**
OWNER: SYSIBM,MYCAT
```

>>>Masks are applied to the source objects before they are compared with the target. You can define as many masks as you want; however, defining many masks will degrade the performance of compare. The first left hand mask that matches are used and the name is translated to the right hand value. If no match is found it is not translated, but still participate in the compare. Using the above masks a

source database with the name 'HLQ47D9' is translated to 'NEW479' before it is compared with the target databases.

For more information about specifying masks, see the information about translation masks and ignore fields in the *DB2 Object Comparison Tool User's Guide*.

> **Default:**
>> blank

compare_mask_name
> The **compare_mask_name** parameter specifies the name of an existing mask entry that is stored in the Change Management database that is to be used for the compare. The **compare_mask_owner** and **compare_mask_name** parameters uniquely identify the mask entry to be used during the compare process. If the compare masks file MASKS DD is pre-allocated and this parameter is specified, the masks that are specified by this parameter are used instead of the pre-allocated compare masks file. The **compare_mask_dsn** and **compare_mask_name** parameters are mutually exclusive.

> **Values:**
>> Specify a valid 1- to 128-character mask name.

> **Default:**
>> blank

compare_mask_owner
> The **compare_mask_owner** parameter specifies the owner of an existing mask entry that is stored in the Change Management database that is to be used for the compare. The **compare_mask_owner** and **compare_mask_name** parameters uniquely identify the mask entry to be used during the compare process.

> **Values:**
>> Specify a valid 1- to 128-character mask owner.

> **Default:**
>> &CURSQLID.

compare_results_comment
> The **compare_results_comment** parameter specifies a comment for the saved compare result. You can use this comment parameter to describe the nature of the compare run. This comment is stored with the saved compare result.

> **Values:**
>> Specify a 1- to 128-character comment or leave this parameter blank.

> **Default:**
>> blank

compare_results_eligible_for_auto_delete
> The **compare_results_eligible_for_auto_delete** parameter specifies when the saved compare result is eligible for deletion by the DB2 Admin's auto-delete process.

> **Values:**

>> **Number of days until eligible for auto-delete**
>>> Specify a number in the range 1-9999.

>> **blank** No auto-deletion will take place.

Default:
blank

compare_results_name

The **compare_results_name** parameter specifies the name for the compare result that is stored in the Change Management database. The **compare_results_owner** and **compare_results_name** together uniquely identify the saved compare result. You can manage the saved compare result using the "MR - Manage saved compare results" dialogs, which you access from the Object Comparison Tool main menu.

Values:

Name for the compare results.
Specify a valid 1- to 128-character compare results name.

Default:
AUTO:&CURTS.

compare_results_owner

The **compare_results_owner** parameter specifies the owner for the compare result that is stored in the Change Management database. The **compare_results_owner** and **compare_results_name** together uniquely identify the saved compare result. You can manage the saved compare result using the "MR - Manage saved compare results" dialogs, which you access from the Object Comparison Tool main menu.

Values:

Owner of the compare result
A valid 1- to 128 character name of the compare results owner.

Default:
&CURSQLID.

content_of_apply_jobs

The **content_of_apply_jobs** parameter specifies whether to generate changes only to database objects and to not generate unloads, loads or other utilities, except REBIND.

Values:

A   Generate all jobs and processes to reload data.

D   Generate only SQL.

Restriction: You must set the **content_of_apply_jobs** parameter to A if the **generate_recover_change** parameter is set to Yes.

Default:
A

data_to_recover

The **data_to_recover** parameter specifies the type of data that the recover change recovers.

Values:

O   Recover using the original data. The original data is the data that is unloaded when the original change is run. If you use the original data during a recovery operation, you might consider whether related tables that were not

affected by the recover also must be restored to the same point to avoid inconsistencies. This option applies only to tables that were dropped in the original change and created in the recover change.

E   Recover using the existing data. If a table is dropped without being re-created in the original change, no data is loaded after the table is created in the recover change.

Default:
E

default_space_priqty

The **default_space_priqty** parameter specifies the default primary space allocation. The default space allocation values are used to allocate, copy, and unload data sets when RUNSTATS or STOSPACE has not been run.

Values:
Specify a valid PRIQTY value.

Default:
30

default_space_secqty

The **default_space_secqty** parameter specifies the default secondary space allocation. The default space allocation values are used to allocate, copy, and unload data sets when RUNSTATS or STOSPACE has not been run.

Values:
Specify a valid SECQTY value.

Default:
30

display_mtc_statements

The **display_mtc_statements** parameter specifies whether the change statements registered for multi-target changes are displayed when analyzing and running of a multi-target change.

Note: These are the change statements on the central system where multi-target change is registered. On each target system, these statements will have variations based on masks applied.

Values:

Y   Display the multi-target change statements

N   Do not display the multi-target change statements.

Default:
N

do_runtime_analyze

The **do_runtime_analyze** parameter specifies whether to do a runtime analyze before a change is run. The runtime analyze is a safety check to ensure a change being run is based on the latest DB2 catalog information.

Values:

Y   Perform a run-time analyze. If the product detects that the latest DB2 catalog information is not used but is needed, the run process will fail with an error. The change will need to be analyzed again before it can be run.

N A runtime analyze is not done before a change is run.

Default:
Y

existing_base_version_action

The **existing_base_version_action** parameter specifies the action to take if a new base version owner and name identify an existing base version.

Values:

REPLACE

The existing base version is replaced with the new base version.

AUTO

The specified base version name is not used. Instead, DB2 Admin uses the product default value for a base version name, such as `AUTO:&CURTS.`. A warning message is issued to notify you of this event.

Default:
AUTO

existing_change_action

The **existing_change_action** parameter specifies the action to be taken when a change already exists.

Values:

REPLACE CONDITIONAL

Replace the change if it is not a prerequisite for other changes. If the change is a prerequisite for other changes, an error message is issued. The names of changes that are dependent on the change are displayed and the replace change request is not processed.

REPLACE UNCONDITIONAL

Replace the change even if it is a prerequisite for other changes. The names of changes that are dependent on the change are displayed.

The change status of changes that are dependent on the change is changed to DEFINED, and the changes must be analyzed before being run.

STOP Do not replace the change.

Default:
STOP

existing_data_set_action

The **existing_data_set_action** parameter specifies the action that occurs if a data set with the same name already exists, and if a supported DB2 Admin data sets is needed. The following data set types support the **existing_data_set_action** parameter:

- CHG sequential file
- IFF PDS
- WSL PDS member
- JCL PDS member
- run job input PDS

If a recover change is generated, the existing data set action option also defines the action for DB2 Admin data sets that are associated with the recover change, for example, recover CHG sequential file or recover IFF PDS, and so on.

Values:

CONDITIONAL
If the data set or PDS member already exists, and the data set or PDS member is already associated with the change from a previous CM action, replace the data set or PDS member. If the data set or PDS member already exists, and the data set or PDS member is not already associated with the change, stop processing.

REPLACE
If the data set or PDS member already exists, replace it.

STOP If the data set or PDS member already exists, stop processing.

Default:
CONDITIONAL

existing_mask_action
The `existing_mask_action` parameter specifies the action that occurs during import mask if the mask specified by `mask_owner` and `mask_owner` parameters identifies an existing mask entry.

Values:

REPLACE CONDITIONAL
Replace the mask if it is not associated with a registered change that needs the mask for implementation. If the mask is needed by one or more changes for implementation, the names of changes are displayed and the mask is not replaced.

REPLACE CONDITIONAL mode does not cover the scenario in which changes can be associated with a mask but the mask is no longer needed to implement the change. For example, when a change is imported using masking, after the import completes the mask is no longer needed to implement that change. The mask, however, is still associated with the change.

REPLACE UNCONDITIONAL
Replace the mask even if it is associated with a registered change that needs the mask for implementation. The names of changes that need the mask for implementation are displayed and the mask is replaced. Only the changes that need the mask for implementation are reported.

REPLACE UNCONDITIONAL mode does not cover the scenario in which changes can be associated with a mask but the mask is no longer needed to implement the change. For example, when a change is imported using masking, after the import completes the mask is no longer needed to implement that change. In this scenario, REPLACE UNCONDITIONAL does not report the imported change. The mask, however, is still associated with the change.

STOP Do not replace the mask.

> **Default:**
> STOP

gen_exclude_name
> The **gen_exclude_name** parameter specifies the name of an Exclude
> Specification that is stored in the Change Management database. The
> Exclude Specification is used for the GEN batch job during the CM batch
> compare.
>
> **Values:**
>
>> A valid 1- to 128-character exclude specification name, blank
>
> **Default:**
> blank

gen_exclude_owner
> The **gen_exclude_owner** parameter specifies the owner of an Exclude
> Specification that is stored in the Change Management database. The
> Exclude Specification is used for the GEN batch job during the CM batch
> compare.
>
> **Values:**
>
>> A valid 1- to 128-character exclude specification owner.
>
> **Default:**
> &CURSQLID

generate_base_version_after_run
> The **generate_base_version_after_run** parameter specifies whether and
> how to automatically generate a new base version after a change is
> implemented. The base version that is generated is associated with the
> change. Automatically generating a base version after a change is run
> enables you to keep a record of object definitions after they were changed,
> and to associate this base version with the change entry.
>
> **Values:**
>
>> **AUTO**
>>> DB2 Admin automatically determines the objects that are
>>> in the base version based on the objects that are being
>>> changed.
>>
>> **USER** The objects that are in the base version are defined by a
>>> version scope that is specified by the user.
>>
>> **NO** A new base version is not generated after the change is
>>> implemented.
>
> **Default:**
> NO

generate_base_version_before_run
> The **generate_base_version_before_run** parameter specifies whether and
> how to automatically generate a new base version before a change is
> implemented. The base version that is generated is associated with the
> change. Automatically generating a base version after a change is run
> enables you to keep a record of object definitions after they were changed
> and to associate this base version with the change entry.
>
> **Values:**

AUTO
DB2 Admin automatically determines the objects that are in the base version based on the objects that are being changed.

USER The objects that are in the base version are defined by a version scope that is specified by the user.

NO A new base version is not generated after the change is implemented.

Default:
NO

generate_job_class
The **generate_job_class** parameter specifies whether to include the **CLASS** parameter on the job card. If you include the **CLASS** parameter on the job card, end the last line of the job card with a comma because DB2 Admin places the **CLASS** parameter on a new line.

Values:

Y Generate a job class parameter with the value of the **job_class** parameter.

N Do not generate a job class parameter.

Default:
Y

generate_recover_change
The **generate_recover_change** parameter specifies whether to generate a recover change if the change does not already have a recover change. If the change already has a recover change, the recover change is regenerated.

Values:

Y A recover change is generated during analyze.

N If the change does not have a recover change, a recover change is not generated. Otherwise, this parameter is forced to be set to Y and the recover change is regenerated.

Default:
N

generate_templates
The **generate_templates** parameter specifies whether to generate templates.

Values:

Y Use the user-defined templates in the ADBTEMPL DD data definition. Refer to Symbol variables in the ADBTEMPL file: DB2 TEMPLATE support for information about using symbol variables to specify DB2 TEMPLATE statements.

N Use the DB2 Admin default template statements.

Default:
N

identity_start_value
The **identity_start_value** parameter specifies the START value of an IDENTITY column of a table if the table is re-created.

Values:

O   The START value from the DB2 catalog is used.

C   The START value is computed based on the identity attributes of the column.

Default:
O

ignore_comment

The **ignore_comment** parameter specifies the comment for a new ignore.

Values:
Specify a 1- 128-character comment or leave this parameter blank.

Default:
blank

ignore_name

The **ignore_name** parameter can be used to specify the name for an existing ignore or a new ignore, depending on what action the Change Management batch interface is invoked. If an ignore is being imported and if the value for the **new_ignore_name** parameter is blank, this parameter specifies the name for the new ignore.

Values:
Specify a valid 1- 128-character ignore name.

Default:
AUTO:&CURTS.

ignore_owner

The **ignore_owner** parameter can be used to specify the owner for an existing ignore or a new ignore, depending on what action the Change Management batch interface is invoked. If an ignore is being imported and if the value for the **new_ignore_owner** parameter is blank, this parameter specifies the owner for the new ignore.

Values:
Specify a valid 1- 128-character ignore owner.

Default:
&CURSQLID.

import_pending_change_action

The **import_pending_change_action** parameter specifies the action that occurs if the import data set contains changes to objects that have changes pending from DB2 Admin Change Management.

Values:

P   Make the pending changes a prerequisite for the imported change.

S   Supersede the pending changes and continue importing the change. The pending changes are placed in DEFINED status and will have the superseded change as a prerequisite.

I   Ignore the pending changes and continue importing the change. Analyzed pending changes are left in ANALYZED status and prerequisites are not established.

C   Cancel the import change process.

Default:
>> P

job_card_line_1
> The **job_card_line_1** parameter specifies line 1 of the job card for generated jobs.
>
> **Values:**
>> Specify a 1- to 72-character statement.
>
> **Default:**
>> //&USERID.D JOB (&SYSUID),'CM BATCH',

job_card_line_2
> The **job_card_line_2** parameter specifies line 2 of the job card for generated jobs.
>
> **Values:**
>> Specify a 1- to 72-character statement.
>
> **Default:**
>> // REGION=0K,NOTIFY=
>> &SYSUID,MSGCLASS=H,MSGLEVEL=(1,1),

job_card_line_3
> The **job_card_line_3** parameter specifies line 3 of the job card for generated jobs.
>
> **Values:**
>> Specify a 1- to 72-character statement.
>
> **Default:**
>> blank

job_card_line_4
> The **job_card_line_4** parameter specifies line 4 of the job card for generated jobs.
>
> **Values:**
>> Specify a 1- to 72-character statement.
>
> **Default:**
>> blank

job_card_line_5
> The **job_card_line_5** parameter specifies line 5 of the job card for generated jobs.
>
> **Values:**
>> Specify a 1- to 72-character statement.
>
> **Default:**
>> blank

job_class
> The **job_class** parameter specifies the CLASS parameter value for the job card.
>
> **Values:**
>> Specify a valid job class.
>
> **Default:**
>> A

job_jcllib_line_1

> The **job_jcllib_line_1** parameter specifies line 1 of the JCLLIB statement. The GOCCM JCL procedure must be accessible in the libraries that are defined by the JCLLIB statement in the run job or in the system procedure libraries.
>
> **Values:**
>> Specify a 1- to 72-character statement.
>
> **Default:**
>> blank
>
> The following example shows how to set this parameter:
>
> ```
> job_jcllib_line_1 = '//GOCCM JCLLIB ORDER=ADB.DEVCUST.JCLLIB'
> ```
>
> This example results in the following JCL line in jobs that are generated by Change Management batch interface:
>
> ```
> //GOCCM JCLLIB ORDER=ADB.DEVCUST.JCLLIB
> ```

job_jcllib_line_2

> The **job_jcllib_line_2** parameter specifies line 2 of the JCLLIB statement.
>
> **Values:**
>> Specify a 1- to 72-character statement.
>
> **Default:**
>> blank

job_jcllib_line_3

> The **job_jcllib_line_3** parameter specifies line 3 of the JCLLIB statement.
>
> **Values:**
>> Specify a 1- to 72-character statement.
>
> **Default:**
>> blank

job_jcllib_line_4

> The **job_jcllib_line_4** parameter specifies line 4 of the JCLLIB statement.
>
> **Values:**
>> Specify a 1- to 72-character statement.
>
> **Default:**
>> blank

job_parm_line_1

> The **job_parm_line_1** parameter specifies line 1 of the job parameter area.
>
> **Values:**
>> Specify a 1- to 72-character statement.
>
> **Default:**
>> blank
>
> The following example shows how to set this parameter:
>
> ```
> JOB_PARM_LINE_1='S=SYS4A'
> ```
>
> This example results in the following line in JCL that is generated by Change Management batch interface:
>
> ```
> /*JOBPARM S=SYS4A
> ```

job_parm_line_2

> The **job_parm_line_2** parameter specifies line 2 of the job parameter area.

Values:
>>Specify a 1- to 72-character statement.

Default:
>>blank

job_parm_line_3
>The **job_parm_line_3** parameter specifies line 3 of the job parameter area.

>>**Values:**
>>>Specify a 1- to 72-character statement.

>>**Default:**
>>>blank

job_parm_line_4
>The **job_parm_line_4** parameter specifies line 4 of the job parameter area.

>>**Values:**
>>>Specify a 1- to 72-character statement.

>>**Default:**
>>>blank

mask_comment
>The **mask_comment** parameter specifies the comment for a new mask.

>>**Values:**
>>>Specify a 1- 128-character comment or leave this parameter blank.

>>**Default:**
>>>blank

mask_ignored_fields
>The **mask_ignored_fields** parameter specifies whether to apply masked values to ignored fields for new (added) objects if the field has been masked and ignored.

>>**Values:**
>>>YES, NO

>>**Default:**
>>>NO

mask_name
>The **mask_name** parameter specifies the name for an existing mask or a new mask, depending on what action the Change Management batch interface is invoked. If a mask is being imported and if the value for the **new_mask_name** parameter is blank, this parameter specifies the name for the new mask.

>>**Values:**
>>>Specify a 1- 128-character mask name or leave this parameter blank.

>>**Default:**
>>>AUTO:&CURTS.

mask_owner
>The **mask_owner** parameter specifies the owner for an existing mask or a new mask, depending on what action the Change Management batch interface is invoked. If a mask is imported and if the value for the **new_mask_owner** parameter is blank, this parameter specifies the owner for the new mask.

Values:

> Specify a 1- 128-character mask owner or leave this parameter blank.

Default:

> &CURSQLID.

max_allocation_to_dasd

> The **max_allocation_to_dasd** parameter specifies the maximum amount of space that can be allocated to DASD. This parameter applies only to new copy and unload data sets. When the space that is required for an unload or copy data set exceeds this threshold value, the data set is allocated to the tape unit that is specified in the next field.
>
> **Values:**
>
> > Specify an integer value.
>
> **Default:**
>
> > 3145680

max_priqty_in_kb

> The **max_priqty_in_kb** parameter specifies the maximum amount of primary space that can be allocated to DASD. This parameter applies only to new copy and unload data sets.
>
> **Values:**
>
> > Specify a valid PRIQTY value. You can specify the following values:
> >
> > * A number that indicates the number of space units specified.
> > * Blank, which causes the kilobyte value shown to be converted to a value that is measured in terms of the space specified.
> > * 99999999, which indicates the maximum space allowed by MVS for the space unit that is specified.
>
> **Default:**
>
> > 3145680

new_base_version_name

> The **new_base_version_name** parameter can be used to specify the default name for a new base version. If this parameter is not blank, this parameter determines the default name for a new base version. Otherwise, the **new_base_version_name** parameter determines the name for a new base version.
>
> If a value is specified for a more specific base version type, for example: **base_version_name_before_run**, that value is used for that base version type instead of the value specified for **new_base_version_name**.
>
> Base version parameter hierarchy:
>
> * base_version_owner
>   * new_base_version_owner
>   - base_version_owner_before_run
>   - base_version_owner_after_run
> * base_version_name
>   * new_base_version_name
>   - base_version_name_before_run
>   - base_version_name_after_run

Values:
>
> 1 to 128 characters

Default:
>
> base_version_name

new_base_version_owner

> The **new_base_version_owner** parameter can be used to specify the default owner for a new base version. If this parameter is not blank, this parameter determines the default name for a new base version. Otherwise, the **new_base_version_name** parameter determines the owner for a new base version.
>
> If a value is specified for a more specific base version type, for example, **base_version_owner_before_run**, that value is used for that base version type instead of the value that is specified for **new_base_version_owner**.
>
> Base version parameter hierarchy:
> * base_version_owner
>   - new_base_version_owner
>   - base_version_owner_before_run
>   - base_version_owner_after_run
> * base_version_name
>   - new_base_version_name
>   - base_version_name_before_run
>   - base_version_name_after_run
>
> **Values:**
> >
> > 1 to 128 characters
>
> **Default:**
> >
> > base_version_owner

new_change_name

> The **new_change_name** parameter can be used to specify the name for a new change. If this parameter is not blank, this parameter determines the name for a new change. Otherwise, the **change_name** parameter determines the name for a new change.
>
> **Values:**
> >
> > Specify a 1- to 128-character change name or leave this parameter blank.
>
> **Default:**
> >
> > blank, which results in the value of the **change_name** parameter being used as the name for the new change.

new_change_owner

> The **new_change_owner** parameter can be used to specify the owner for a new change. If this parameter is not blank, this parameter determines the owner for a new change. Otherwise, the **change_owner** parameter determines the owner for a new change.
>
> **Values:**
> >
> > Specify a 1- to 128-character change owner or leave this parameter blank.
>
> **Default:**
> >
> > blank, which results in the value of the **change_owner** parameter being used as the name for the new change owner.

new_ignore_name

The **new_ignore_name** parameter can be used to specify the name for a new ignore. If this parameter is not blank, it determines the name for a new ignore. Otherwise, the **ignore_name** parameter determines the name for a new ignore.

Values:

Specify a 1- to 128-character ignore name or leave this parameter blank.

Default:

blank, which results in the value of the **ignore_name** parameter being used as the name for the new ignore.

new_ignore_owner

The **new_ignore_owner** parameter can be used to specify the owner for a new ignore. If this parameter is not blank, it determines the owner for a new ignore. Otherwise, the **ignore_owner** parameter determines the owner for a new ignore.

Values:

Specify a 1- to 128-character ignore owner or leave this parameter blank.

Default:

blank, which results in the value of the **ignore_owner** parameter being used as the name for the new ignore owner.

new_mask_name

The **new_mask_name** parameter can be used to specify the name for a new mask. If this parameter is not blank, it determines the name for a new mask. Otherwise, the **mask_name** parameter determines the name for a new mask.

Values:

Specify a 1- to 128-character mask name or leave this parameter blank.

Default:

blank, which results in the value of the **mask_name** parameter being used as the name for the new mask name.

new_mask_owner

The **new_mask_owner** parameter can be used to specify the owner for a new mask. If this parameter is not blank, it determines the owner for a new mask. Otherwise, the **mask_owner** parameter determines the owner for a new mask.

Values:

Specify a 1- to 128-character mask owner or leave this parameter blank.

Default:

blank, which results in the value of the **mask_owner** parameter being used as the name for the new mask owner.

disable_optimize_reorg

The **disable_optimize_reorg** parameter specifies whether the compare process should disable the optimization of REORG statements.

Values:

Y   Compare disables the optimization of REORG statements.

<table>
<tr><td></td><td>N</td><td>Compare does not disable the optimization of REORG statements.</td></tr>
</table>

ovr_configdb_error

The **ovr_configdb_error** parameter specifies whether DB2 Admin should continue processing when change information is unable to be stored in the InfoSphere® Optim Configuration Manager repository database or the backup tables on the local system. This option applies only if integration with InfoSphere Optim Configuration Manager is enabled and the action on error setting is set to allow the override parameter.

Values:

YES   If integration with InfoSphere Optim Configuration Manager (OCM) is enabled and the action on error setting is set to allow the override parameter, DB2 Admin will continue processing the change even if the OCM repository database and the backup tables on the local system are not available.

NO   If integration with InfoSphere Optim Configuration Manager (OCM) is enabled, DB2 Admin will stop processing the change if the OCM repository database and the backup tables on the local system are not available.

Default:
NO

pds_for_recover_jcl

The **pds_for_recover_jcl** parameter specifies the name of a PDS to store the generated recover jobs.

Values:

A valid PDS data set name
Specify a 1- to 46-character data set name. If this name is not enclosed in single quotation marks, the fully qualified data set name is **prefix_for_data_sets**.**pds_for_recover_jcl**.

Default:
&**SSID**..RECOVER.JCL

pds_for_recover_wsl

The **pds_for_recover_wsl** parameter specifies the name of a PDS to store the work statement lists (WSLs) that the analyze job generates for the recover change.

Values:

A valid PDS data set name
Specify a 1- to 46-character data set name. If this name is not enclosed in single quotation marks, the fully qualified data set name is **prefix_for_data_sets**.**pds_for_recover_wsl**.

Default:
&**SSID**..RECOVER.WSL

pds_for_run_jcl

The **pds_for_run_jcl** parameter specifies the name of a PDS to store the generated run jobs.

Values:

> **A valid PDS data set name**
>> Specify a 1- to 46-character data set name. If this name is not enclosed in single quotation marks, the fully qualified data set name is **prefix_for_data_sets**.**pds_for_run_jcl**.

Default:
> &**SSID**..RUN.JCL

pds_for_run_job_input
> The **pds_for_run_job_input** parameter specifies the name of a PDS in which the run job or recover job input data is stored. This parameter is used only when **use_permanent_data_set_for_run_job_input** is set to Y. You must ensure the same run job input PDS is not used for different changes. Using the same run job input PDS for different changes can cause problems when a change is run.

> **Values:**

>> **A valid PDS data set name**
>>> Specify a 1- to 46-character data set name. If this name is not enclosed in single quotation marks, the fully qualified data set name is **prefix_for_data_sets**.**pds_for_run_job_input**.

> **Default:**
>> &**SSID**..&CHGTAG..IN

pds_for_wsl
> The **pds_for_wsl** parameter specifies the name of the PDS to store the work statement list (WSL) that the analyze job generates for the change.

> **Values:**

>> **A valid PDS data set name**
>>> Specify a 1- to 46-character data set name. If this name is not enclosed in single quotation marks,, the fully qualified data set name is **prefix_for_data_sets**.**pds_for_wsl**.

> **Default:**
>> &**SSID**..RUN.WSL

percent_increase_for_converted_data_sets
> The **percent_increase_for_converted_data_sets** parameter specifies the percentage increase in size of the converted unload data set over the unload data set. The ALT/Object Compare process converts data from the UNLOAD step. The newly converted data might require more space than the unload data set. This parameter allows you to increase the size of the converted data set by a percentage greater than the unload data set, therefore helping to avoid out-of-space conditions.

> **Values:**
>> A number in the range 0-100.

> **Default:**
>> 0

plan   The **plan** parameter specifies the DB2 plan name to connect with.

> **Values:**
>> Specify a 1- to 8-character DB2 plan name.

Default:
　　ADB

prefix_for_data_sets
　　The **prefix_for_data_sets** parameter specifies the data set prefix that is
　　used when data sets are allocated, such as: WSL PDS, JCL PDS, UNLOAD,
　　LOAD, and so on.

　　Values:
　　　　Specify a 1- to 17-character data set prefix.

　　Default:
　　　　&USERID.

processing_order
　　The **processing_order** parameter specifies the order that the objects will be
　　processed in.

　　Values:

　　　　T　　Object type processing will be performed. This process
　　　　　　compares one object type at a time.

　　　　H　　Database hierarchy processing will be performed. This
　　　　　　process compares all object types within a database
　　　　　　hierarchy at the same time. This is the default value.

　　　　Note: If you choose not to disable REORG optimization, the
　　　　compare process will set your processing_order value to H and
　　　　continue processing using REORG optimization.

　　Default:
　　　　H

recover_change_comment
　　The **recover_change_comment** parameter specifies the comment for a
　　recover change.

　　Values:
　　　　Specify a 1- to 128-character comment or leave this field blank.

　　Default:
　　　　blank

recover_change_name
　　The **recover_change_name** parameter can be used to specify the name for a
　　new recover change. If this parameter is not blank, this parameter
　　determines the name for a new recover change. Otherwise, the name for a
　　new recover change is the original change name with _RCVR appended.

　　Values:
　　　　Specify a 1- to 128-character change name.

　　Default:
　　　　The name of original change with _RCVR appended

recover_change_owner
　　The **recover_change_owner** parameter can be used to specify the owner for
　　a new recover change. If this parameter is not blank, this parameter
　　determines the owner for a new recover change. Otherwise, the owner for
　　a new recover change is the same owner as its original change.

　　Values:
　　　　Specify a 1- to 128-character change owner.

Default:
> The owner of original change

recover_pending_change_action
> The `recover_pending_change_action` parameter specifies the action that occurs if the change being recovered contains changes to objects that have changes pending from DB2 Admin Change Management.
>
> **Values:**
>
> > **S** This option recovers the specified change and set to DEFINED status for any pending change that modifies the same or related objects. The recover change supersedes any pending changes that modify the same or related objects.
> >
> > **C** This option prevents the change from being recovered when pending changes will modify the same or related objects. If there are pending changes, the changes are not recovered. To recover this change and to set the status of any pending changes to DEFINED, set the value of this parameter to S (supersede).
>
> **Default:**
> > C

reload_accelerated_tables
> The `reload_accelerated_tables` parameter specifies whether to automatically detect and reload accelerated tables in situations of DROP/CREATE or ALTER of accelerated tables in DB2, involving changes to its definition, data or partitions.
>
> **Y** Reload accelerated tables.
>
> **N** Do not reload accelerated tables.
>
> **Default:**
> > Y

report_expected_conversion_problems
> The `report_expected_conversion_problems` parameter specifies whether a report is generated of the data conversion problems for tables that are expected to occur when the change is run.
>
> **Values:**
>
> > **Y** The report includes the expected conversion problems for tables when the change is run.
> >
> > **N** The report does not include a list of expected conversion problems.
>
> **Default:**
> > N

report_object_count
> The `report_object_count` parameter specifies whether a statistics report is generated of compared and changed objects for each object type.
>
> **Values:**
>
> > **Y** The report includes statistics of compared and changed objects for each object type.
> >
> > **N** The report does not include the object count statistics.

Default:
> N

report_object_totals
> The **report_object_totals** parameter specifies whether a summary report of overall totals for each object type is generated. The number of altered, created, or dropped objects for each object type is included.
>
> The summary report contains the overall totals of objects affected and includes values from the analyze step from each of the target systems.
>
> **Note:** Specifying report_object_totals = 'Y' forces report_object_counts to be set to 'Y'.
>
> **Values:**
>
> > **Y** The report includes the object totals report.
> >
> > **N** The report does not include the object totals report.
>
> **Default:**
> > N

report_only_changed_objects
> The **report_only_changed_objects** parameter specifies whether to report objects which are identical in the source and the target.
>
> **Values:**
>
> > **Y** The report does not include objects that are identical in the source and target.
> >
> > **N** The report includes objects that are identical in the source and the target.
>
> **Default:**
> > N

report_summary
> The **report_summary** parameter specifies whether to include a brief summary of changes for each object in the report.
>
> **Values:**
>
> > **Y** The report includes a brief summary of changes for each object.
> >
> > **N** The report does not include a brief summary.
>
> **Default:**
> > N

report_system_generated_ignore_fields
> The **report_system_generated_ignore_fields** parameter specifies whether to include in the report the system generated names of the fields that are ignored.
>
> **Values:**
>
> > **Y** The report includes system generated names of the fields that are ignored by the compare or analyze process.
> >
> > **N** The report does not include system generated names of the fields that are ignored.

Default:
>   N

report_translation_masks
>   The **report_translation_masks** parameter specifies whether to report the
>   translation masks that are used.
>
>   **Values:**
>   >   Y, N
>   >
>   >   **Y**   The report includes the masks used by the compare or
>   >   analyze process.
>   >
>   >   **N**   The report does not include the masks that are used.
>
>   **Default:**
>   >   N

report_user_specified_ignore_fields
>   The **report_user_specified_ignore_fields** parameter specifies whether
>   the report includes the user-defined names of the fields that are ignored.
>
>   **Values:**
>   >   **Y**   The report includes user-defined names of the fields that
>   >   are ignored by the compare or analyze process.
>   >
>   >   **N**   The report does not include user-defined names of the
>   >   fields that are ignored.
>
>   **Default:**
>   >   N

retain_generated_always_for_row_change_ts
>   The **retain_generated_always_for_row_change_ts** parameter specifies
>   whether to retain GENERATED ALWAYS for the designated column types
>   (ROWID or ROW CHANGE TIMESTAMP).
>
>   **Values:**
>   >   **Y**   Retain the GENERATED ALWAYS attribute for row change
>   >   time stamp columns.
>   >
>   >   **N**   Do not retain the GENERATED ALWAYS attribute for row
>   >   change time stamp columns.
>
>   **Default:**
>   >   N

retain_generated_always_for_rowid
>   The **retain_generated_always_for_rowid** parameter specifies whether to
>   retain GENERATED ALWAYS for the designated column types (ROWID or
>   ROW CHANGE TIMESTAMP).
>
>   **Values:**
>   >   **Y**   Retain the GENERATED ALWAYS attribute for rowid
>   >   columns.
>   >
>   >   **N**   Do not retain the GENERATED ALWAYS attribute for
>   >   rowid columns.
>
>   **Default:**
>   >   N

run_check_data

The **run_check_data** parameter specifies whether to generate a CHECK DATA utility job for the table spaces that are affected by the (RE)LOAD utility jobs that the analyze process generates in the WSL.

Values:

    Y   Generate a CHECK DATA utility job for each table space that is affected by a LOAD utility.

    N   Do not generate a CHECK DATA utility job.

Default:
    N

run_rebind

The **run_rebind** parameter specifies whether to generate a job to rebind plans and packages that are affected by changes that the analyze process generates.

Values:

    Y   Generate REBIND statements for packages and plans that are affected by the change.

    N   Do not generate REBIND utility statements.

Default:
    N

run_reorg_rebuild

The **run_reorg_rebuild** parameter specifies whether to generate REORG table space and REBUILD index utility jobs after applying the changes from the analyze process, the purpose of which is to make the target operational.

Values:
    M, A, N

    M   Mandatory. Generate REORG utility statements to remove REORG pending conditions.

    A   All relevant. Generate all needed REORG utility statements to fully implement the effects of the changes, for example, space parameter changes.

    N   None. No REORG utility statements are generated. This option is invalid if you specified No to Allow rotate parts.

Default:
    N

run_runstats

The **run_runstats** parameter specifies whether to generate a RUNSTATS utility job for the table spaces that are affected by the RE(LOAD) utility jobs and for the table spaces, tables, and indexes that are affected by SQL ALTER statements that the analyze process generates in the WSL.

Values:

    R   Generate RUNSTATS utility statements for all tables that are affected by the (RE)LOAD utility.

    A   Generate RUNSTATS utility statements for all altered table space, table, and index objects.

B   Generate RUNSTATS utility statements for objects that are affected by the RE(LOAD) utility and SQL ALTER statements.

N   No RUNSTATS utility statements are generated.

Default:
>N

run_sqlid
>The **run_sqlid** parameter specifies whether SET CURRENT SQLID statements are generated and, if so, what SQLID value to use.

>**Values:**

>>**An SQLID**
>>>The specified Run SQLID is the owner of databases and table spaces. If the specified Run SQLID is different from the current owner, the databases, table spaces, and all dependent objects are dropped and re-created to accomplish the change of owner.

>>**<NONE>**
>>>No SET CURRENT SQLID statements are generated.

>>**blank** SET CURRENT SQLID statements are generated when necessary.

>**Default:**
>>blank

save_compare_results
>The **save_compare_results** parameter specifies whether compare results are saved during the compare run. You can manage the saved compare result using the "MR - Manage saved compare results" dialogs, which you access from the Object Comparison Tool main menu.

>**Values:**
>>YES, NO

>**Default:**
>>NO

save_source_base_version
>The **save_source_base_version** parameter specifies whether to save the source base version that is generated for the change during the analyze process. The source base version represents the DB2 object definitions after the change is implemented.

>**Values:**

>>**Y**   The source base version generated during analyze is saved as a new base version.

>>**N**   The source base version generated during analyze is not saved.

>**Default:**
>>N

save_target_base_version
>The **save_target_base_version** parameter specifies whether to save the target base version that is generated for the change during the analyze process. The target base version represents the DB2 object definitions as

they existed in the DB2 catalog at analyze time, with DB2 Admin change management pending changes applied, but without the changes for the specified change applied.

Values:

    Y   The target base version that was generated during analyze is saved as a new base version.

    N   The source base version during analyze is not saved.

Default:
    N

sequence_restart_value

The **sequence_restart_value** parameter specifies what the value for the RESTART attribute is when a DB2 sequence object is re-created. Use this parameter only for recovery paths.

Values:
    ORIGINAL, COMPUTED

Default:
    ORIGINAL

source_dsn

The **source_dsn** parameter specifies the name of the data set that contains the compare source. Specifying this parameter overrides a pre-allocated compare source input file (SRCIN DD).

Values:

    A data set name
        Specify a 1- to 46-character data set name. If **source_type** = 'DDL', specify the name of the data set that contains the DDL for the compare source.

        If **source_type** = 'USER', specify the name of the data set that contains the list of DB2 Admin quick scopes for the compare source.

    blank If **source_type** = 'DDL', the SRCIN file must contain the DDL for the compare source.

        If **source_type** = 'USER', either the **source_version_scope_owner** and **source_version_scope_name** parameters must be specified, or the SRCIN file must contain the list of DB2 Admin quick scopes for the compare source.

Default:
    blank

source_exclude_name

The **source_exclude_name** parameter specifies the name of an Exclude Specification that is stored in the Change Management database. The **source_exclude_owner** and **source_exclude_name** parameters identify an existing Exclude Specification to be used for the compare source.

Values:
    A valid exclude specification name, blank

A valid exclude specification name.
Specify a 1- to 128-character exclude specification name.
The specified Exclude Specification is used for the source
during the compare process.

blank   Exclude objects are not used for the compare source.

Default:
blank

source_exclude_owner
The **source_exclude_owner** parameter specifies the owner of an Exclude
Specification that is stored in the Change Management database. The
source_exclude_owner and **source_exclude_name** parameters identify an
existing Exclude Specification to be used for the compare source.

Values:
Specify a valid 1- to 128-character exclude specification owner.

Default:
&CURSQLID.

source_location
The **source_location** parameter specifies the DB2 location for the compare
source when the DB2 objects are located in a DB2 subsystem.

Values:
Specify a valid 1- to 128-character location name that is defined in
SYSIBM.LOCATIONS or leave this parameter blank to specify the
local DB2 subsystem.

blank   The local DB2 subsystem.

Default:
blank

source_type
The **source_type** parameter specifies the type of input that identifies the
DB2 objects for the source of the compare.

Values:

DDL   The source is DDL. You can use the compare source input
file (SRCIN DD) or the **source_dsn** parameter to specify a
data set that contains the DDL. If the **source_dsn** parameter
is not specified, the compare source input file (SRCIN DD)
must be pre-allocated.

USER   The source is a DB2 subsystem and the list of object names
is provided by the user. you can use a DB2 Admin version
scope, a list of DB2 Admin quick scopes, or both, to specify
the list of DB2 objects for the compare source.

The **source_version_scope_owner** and
source_version_scope_name parameters specify an existing
version scope. The compare source input file (SRCIN DD)
or the **source_dsn** parameter can be used to specify a data
set that contains a list of DB2 Admin quick scopes.

Refer to "Version scopes" on page 763 for information
about using DB2 Admin quick scopes to specify DB2
objects.

Default:
>> DDL

source_version_comment
> The **source_version_comment** parameter specifies a comment or description of the source version.

> **Values:**
>> Specify a 1- to 128-character comment, or leave this field blank.

> **Default:**
>> blank

source_version_name
> The **source_version_name** parameter specifies the name for the base version that will store the generated source base version work file. If the **source_version_owner** and **source_version_name** parameters identify an existing base version, the **existing_base_version_action** parameter controls whether the existing base version is replaced or a product-generated version name is used.

> **Values:**
>> Specify a valid 1- to 128-character version name.

> **Default:**
>> AUTO:OC.&CURTS..SRCVF

source_version_owner
> The **source_version_owner** parameter specifies the owner for the base version that will store the generated source base version work file. If the **source_version_owner** and **source_version_name** parameters identify an existing base version, the **existing_base_version_action** parameter controls whether the existing base version is replaced or a product-generated version name is used.

> **Values:**
>> Specify a valid 1- to 128-character version owner.

> **Default:**
>> &CURSQLID.

source_version_scope_name
> The **source_version_scope_name** parameter specifies the name of the version scope for the compare source. It is only used if the source type is USER.

> **Values:**
>> A valid version scope name; 1 to 128 characters, blank

>> **A valid version scope name.**
>>> Specify a valid 1- to 128-character version scope name. If source_type = 'USER', the version scope that is specified by **source_version_scope_owner** and **source_version_scope_name** is used for the DB2 object list for the compare source.

>> **blank** If source_type = 'USER', a list of DB2 Admin quick scopes must be specified in a pre-allocated SRCIN DD file or in the data set that is specified by the **source_dsn** parameter.

> **Default:**
>> blank

source_version_scope_owner

> The **source_version_scope_owner** parameter specifies the owner of the version scope for the compare source. This parameter is used only if the source type is USER.
>
> **Values:**
> > Specify a valid 1- to 128-character version scope owner.
>
> **Default:**
> > &CURSQLID.

source_version_type

> The **source_version_type** parameter specifies the final disposition of the generated source base version work file. If the SRCVF file is pre-allocated, this parameter has no effect for types FILE and TEMP.
>
> **Values:**
>
> > **FILE** If the SRCVF file is not pre-allocated, it is allocated by using the attributes from **admin_dataset_type** = 'SRCVF'.
> >
> > **DB2** If the SRCVF file is not pre-allocated, it is allocated by using the attributes from **admin_dataset_type** = 'SRCVF' but as a temporary file. The file contents are stored in the DB2 Admin change management repository using the owner and name values from the **source_version_owner** and **source_version_name** parameters.
> >
> > **TEMP** If the SRCVF file is not pre-allocated, it is allocated by using the attributes from **admin_dataset_type** = 'SRCVF' but as a temporary file.
>
> **Default:**
> > FILE

space_tape_unit

> The **space_tape_unit** parameter specifies the name of a valid tape unit. This parameter applies only to new copy and unload data sets.
>
> **Values:**
> > Specify a valid space unit for tape.
>
> **Default:**
> > TAPE

space_unit

> The **space_unit** parameter specifies the units in which new data sets are to be allocated. This parameter applies only to new copy and unload data sets. Specifying BLK causes DB2 Admin to allocate in blocks of 8192 bytes, which is the block size used by the DB2 Unload utility.
>
> **Values:**
> > Specify a valid space unit.
>
> **Default:**
> > TRK

space_unit_name

> The **space_unit_name** parameter specifies the default unit name.
>
> **Values:**
> > Specify a valid space unit name.

Default:
> SYSALLDA

ssid The **ssid** parameter specifies the DB2 subsystem to connect to.

> **Values:**
> > Specify a valid 1- to 4-character DB2 subsystem ID.
>
> **Default:**
> > This parameter does not have a default value.

stop_on_conversion_error
> The **stop_on_conversion_error** parameter specifies whether to stop WSL processing when data conversion errors occur.
>
> **Values:**
>
> > **Y** Stop WSL processing with RC=28 when conversion errors occur.
> >
> > **N** Do not stop WSL processing when conversion errors occur.
>
> **Default:**
> > N

suppress_adding_columns
> The **suppress_adding_columns** parameter specifies whether compare should suppress adding target columns.
>
> **Values:**
> > YES, NO
>
> **Default:**
> > NO

suppress_drop_of_columns
> The **suppress_drop_of_columns** parameter specifies whether compare should suppress dropping target columns.
>
> **Values:**
> > YES, NO
>
> **Default:**
> > NO

suppress_drop_of_objects
> The **suppress_drop_of_objects** parameter specifies whether the compare process will suppress dropping target objects that are in the target but that are not in the source.
>
> **Values:**
> > YES, NO
>
> **Default:**
> > NO
>
> Regardless of the value that you set for this option, DB2 Object Comparison Tool replaces all relationships between a parent and a child if a foreign key is specified in the source. To delete a foreign key, both the parent and the child must be present in the source (without a foreign key). If DROP statements are part of the source DDL, objects are dropped regardless of the value that is specified for this parameter.
>
> Regardless of the value that you set for this option, DB2 Object Comparison Tool drops all explicit LOB objects from the target if they are

not specified on the source. However, if the base table that is associated with the LOB objects is kept because 'Suppress DROP of objects' is set to 'YES', then all of the LOB objects are kept.

Note: If the target_type = 'AUTO' for Target is used, the `suppress_drop_of_objects` parameter is forced to a setting of YES. If NO was specified, a warning message is issued stating that the change was made.

symbol_name
> The `symbol_name` parameter specifies the name of a user-defined symbol variable to use to mask some of the parameter values at run time.
>
> **Values:**
>
> > **a valid symbol variable name**
> > > Specify a valid symbol variable name or leave this parameter blank. A valid symbol variable name begins with the ampersand (&) character and ends with the **.** character. The name can be 3-128 characters, the total of which includes the & and **.** characters. The name is converted to upper case.
>
> **Default:**
> > blank

symbol_value
> The `symbol_value` parameter specifies the value of a user-defined symbol variable to be used to mask some of the parameter values at run time.
>
> **Values:**
> > Specify a 1- to 128-character value or leave this field blank.
>
> **Default:**
> > blank

take_an_image_copy
> The `take_an_image_copy` parameter specifies whether to generate a COPY utility job for the table spaces that are affected by the RE(LOAD) utility jobs and for the table spaces, tables, and indexes that are affected by SQL ALTER statements that the analyze process generates in the WSL.
>
> **Values:**
>
> > **R**   Generate COPY utility statements for all tables that are affected by the (RE)LOAD utility.
> >
> > **A**   Generate COPY utility statements for all altered table space, table, and index objects.
> >
> > **B**   Generate COPY utility statements for objects that are affected by the RE(LOAD) utility and SQL ALTER statements.
> >
> > **N**   No COPY utility statements are generated.
>
> **Default:**
> > N

target_associationID
> The `target_associationID` parameter specifies the association ID provided by multi-target central system used to identify the target change.

Values

>The value originates from the multi-target change file, which cannot be modified by the user.

Default:

>Blank

target_change_comment

>The **target_change_comment** parameter specifies the comment for a new change on the target system.

>**Values:**

>>1 to 128 characters

>**Default:**

>>blank

target_dsn

>The **target_dsn** parameter specifies the name of the data set that contains the compare target. This parameter is used when the target_type is USER. Specifying this parameter overrides a pre-allocated compare target input file (TGTIN DD).

>**Values:**

>>**A data set name.**

>>>Specify a 1- to 46-character data set name. If target_type = 'USER', specify the name of the data set that contains the list of DB2 Admin quick scopes for the compare target; One to 46 characters

>>**blank** If target_type = 'USER', either the **target_version_scope_owner** and **target_version_scope_name** parameters must be specified, or the TGTIN file must contain the list of DB2 Admin quick scopes for the compare target.

>**Default:**

>>blank

target_exclude_name

>The **target_exclude_name** parameter specifies the name of an Exclude Specification that is stored in the Change Management database. The **target_exclude_owner** and **target_exclude_name** parameters identify an existing Exclude Specification to be used for the compare target.

>**Values:**

>>**A valid exclude specification name**

>>>Specify a valid 1- to 128-character exclude specification name. The specified Exclude Specification is used for the target during the compare process. One to 128 characters

>>**blank** Exclude objects is not used for the compare target.

target_exclude_owner

>The **target_exclude_owner** parameter specifies the owner of an Exclude Specification that is stored in the Change Management database. The **target_exclude_owner** and **target_exclude_name** parameters identify an existing Exclude Specification to be used for the compare target.

>**Values:**

>>Specify a valid 1- to 128-character exclude specification owner.

Default:
    &CURSQLID.

target_ignore_name

The **target_ignore_name** parameter specifies the name of an existing Ignore Fields entry as defined in the Change Management database on the target system. The Ignore Fields entry on the target system, that is identified by the **target_ignore_owner** and **target_ignore_name** parameters, is used to ignore the DB2 columns when the change on the target system is analyzed.

Values:

    A valid Ignore Fields name
        Specify a valid 1- to 128-character ignore fields name.

    blank Ignore Fields name is not included in the statement.

Default:
    blank

target_ignore_owner

The **target_ignore_owner** parameter specifies the owner of an existing Ignore fields entry defined in the Change Management database on the target system. The Ignore Fields entry on the target system, that is identified by the **target_ignore_owner** and **target_ignore_name** parameters, is used to ignore the DB2 columns when the change on the target system is analyzed.

Values:

    A valid Ignore Fields owner
        Specify a valid 1- to 128-character target ignore owner.

    blank Ignore Fields owner is not included in the statement.

Default:
    blank

target_location

The **target_location** parameter specifies the DB2 location for the compare target when the DB2 objects are located in a DB2 subsystem. If the compare result is imported as a new change (**action_import_change** = 'Y') the target location must be the local DB2 subsystem.

Values:
    Specify a location that is defined in SYSIBM.LOCATIONS or leave this field blank to specify the local DB2 subsystem.

Default:
    blank

target_mask_name

The **target_mask_name** parameter specifies the name of an existing mask defined in the Change Management database on the target system. The mask on the target system, that is identified by the **target_mask_owner** and **target_mask_name** parameters, is used to mask the change statements when the change on the target system is registered.

Values:

    A valid mask name
        Specify a valid 1- to 128-character target mask name.

    blank The mask name is not included in the statement.

Default:
blank

target_mask_owner

The **target_mask_owner** specifies the owner of an existing mask defined in the Change Management database on the target system. The mask on the target system, that is identified by the **target_mask_owner** and **target_mask_name** parameters, is used to mask the change statements when the change on the target system is registered.

Values:

A valid mask owner
Specify a valid 1- to 128-character target mask owner.

blank   The mask owner is not included in the statement.

Default:
blank

target_type

The **target_type** parameter specifies the type of input that identifies the DB2 objects for the target of the compare.

Values:
AUTO, USER

AUTO
The target is a DB2 subsystem. The DB2 objects for the compare target are automatically selected by the product based on the content of the compare source.

USER   The target is a DB2 subsystem and the list of object names is provided by the user. You can use a DB2 Admin version scope, a list of DB2 Admin quick scopes, or both, to specify the list of DB2 objects for the compare target.

The **target_version_scope_owner** and **target_version_scope_name** parameters specify an existing version scope. The compare target input file (TGTIN DD) or the **target_dsn** parameter can be used to specify a data set that contains a list of DB2 Admin quick scopes.

Refer to "Version scopes" on page 763 for information about how to specify the DB2 objects using DB2 Admin quick scopes to define DB2 objects..

Default:
AUTO

target_version_comment

The **target_version_comment** parameter specifies a comment or description of the target version.

Values:
Specify a 1- to 128-character comment or leave this field blank.

Default:
blank

target_version_name

The **target_version_name** parameter specifies the name for the base version that will store the generated target base version work file. If the **target_version_owner** and **target_version_name** parameters identify an

existing base version, the **existing_base_version_action** parameter controls whether the existing base version is replaced or a product-generated version name is used.

Values:
> Specify a valid 1- to 128-character version name.

Default:
> AUTO:OC.&CURTS..TGTVF

target_version_owner
> The **target_version_owner** parameter specifies the owner for the base version that will store the generated target base version work file. If the **target_version_owner** and **target_version_name** parameters identify an existing base version, the **existing_base_version_action** parameter controls whether the existing base version is replaced or a product-generated version name is used.
>
> **Values:**
>> Specify a valid 1- to 128-character version owner.
>
> **Default:**
>> &CURSQLID.

target_version_scope_name
> The **target_version_scope_name** parameter specifies the name of the version scope for the compare target. It is only used if the target type is USER.
>
> **Values:**
>
>> **A valid version scope name.**
>>> Specify a valid 1- to 128-character version scope name. If target_type = 'USER', the version scope that is specified by **target_version_scope_owner** and **target_version_scope_name** is used for the DB2 object list for the compare target.
>>
>> **blank** If **target_type = 'USER'**, a list of DB2 Admin quick scopes must be specified in a pre-allocated TGTIN DD file or in the data set specified by the **target_dsn** parameter.
>
> **Default:**
>> blank

target_version_scope_owner
> The **target_version_scope_owner** parameter specifies the owner of the version scope for the compare target. This parameter is used only if the target_type parameter is set to USER.
>
> **Values:**
>> Specify a valid 1- to 128-character version scope owner.
>
> **Default:**
>> &CURSQLID.

target_version_type
> The **target_version_type** parameter specifies the final disposition of the generated target base version work file. If the TGTVF file is pre-allocated this parameter has no effect for types FILE and TEMP.
>
> **Values:**

FILE If the TGTVF file is not pre-allocated, it is allocated by using the attributes from **admin_dataset_type** = 'TGTVF'.

DB2 If the TGTVF file is not pre-allocated, it is allocated by using the attributes from **admin_dataset_type** = 'TGTVF' but as a temporary file. The file contents are stored in the DB2 Admin change management repository. The owner and name values are obtained from the **target_version_owner** and **target_version_name** parameters.

TEMP If the TGTVF file is not pre-allocated, it is allocated by using the attributes from **admin_dataset_type** = 'TGTVF', but as a temporary file.

Default:
FILE

unload_method

The **unload_method** parameter specifies the method that is used to unload the data.

Values:

U Use the UNLOAD utility.

P Use the DB2 Parallel UNLOAD utility.

H Use DB2 High Performance Unload for z/OS (HPU) when available. The HPU option is supported only if an HPU load library is specified.

Default:
U

use_defer_yes

The **use_defer_yes** parameter specifies whether to use DEFER YES clauses on any eligible CREATE INDEX statements. Any user-specified masks will have precedence. This value is also used for subsequent runtime analysis to ensure that the same DDL and DB2 Admin statements are generated.

Values:

Y Specify DEFER YES on eligible indexes.

N Do not specify DEFER YES.

Default:

use_ignore_for_import_change

The **use_ignore_for_import_change** parameter specifies whether an ignore is used for the imported change.

Values:
Y, N, blank

Y If an ignore is also being imported, the ignore that is used for import change is the newly created ignore. Otherwise, the ignore that is used is identified by the **ignore_owner** and **ignore_name** parameters.

blank Specifies that this parameter defaults to Y if an ignore and a change are imported.

Default:
> blank

use_mask_for_export_change
> The **use_mask_for_export_change** parameter specifies whether the data for export change is masked during export.

> **Values:**
>> Y, N, blank

>> **Y**   If a mask is also being exported, the mask that is used for export change is the newly created mask. Otherwise, the mask that is used is identified by the **mask_owner** and **mask_name** parameters.

>> **blank**   Specifies that this parameter defaults to Y if a mask and a change are imported.

> **Default:**
>> blank

use_mask_for_import_change
> The **use_mask_for_import_change** parameter specifies whether the input for import change is masked during import.

> **Values:**
>> Y, N, blank

>> **Y**   If a mask is also being imported, the mask that is used for import change is the newly created mask. Otherwise, the mask that is used is identified by the **mask_owner** and **mask_name** parameters.

>> **blank**   Specifies that this parameter defaults to Y if a mask and a change are imported.

> **Default:**
>> blank

use_permanent_data_set_for_run_job_input
> The **use_permanent_data_set_for_run_job_input** parameter specifies where to store the run job input. The run job input can be put in-stream in the run job itself, or into a PDS.

> **Values:**

>> **Y**   Store the run job input data in a permanent data set that is referenced in the run job.

>> **N**   Store the run job input data in an in-stream data set in the run job.

> **Default:**
>> N

use_utility_options
> The **use_utility_options** parameter specifies whether to use the customized utility options.

> **Values:**

>> **Y**   The user-customized utility options are used.

>> **N**   The DB2 Admin and DB2 default utility options are used.

Default:
>N

util_check_auxerror
>The **util_check_auxerror** parameter specifies the AUXERROR option for generated CHECK DATA utility statements.

>**Values:**

>>**R** AUXERROR REPORT is added.

>>**I** AUXERROR INVALIDATE is added.

>>**blank** The AUXERROR option is not added; DB2 default utility options are used.

>**Default:**
>>blank

util_check_drain_wait
>The **util_check_drain_wait** parameter specifies the DRAIN_WAIT option for generated CHECK DATA utility statements.

>**Values:**

>>**A valid DRAIN_WAIT value for CHECK DATA; 1 - 1800**
>>>Specify a DRAIN_WAIT setting in the range 1 - 1800. The DRAIN_WAIT option is added with the specified value.

>>**blank** The option is not added to the utility statement; DB2 default utility options are used.

>**Default:**
>>blank

util_check_exceptions
>The **util_check_exceptions** parameter specifies the EXCEPTIONS option for generated CHECK DATA utility statements.

>**Values:**

>>**A valid EXCEPTIONS value for CHECK DATA)**
>>>Specify a valid EXCEPTIONS value in the range 0 - 32767. The EXCEPTIONS option is added with the specified value, for example: EXCEPTIONS 2

>>**blank** The option is not added to the utility statement; DB2 default utility options are used.

>**Default:**
>>blank

util_check_include_xml_tablespaces
>The **util_check_include_xml_tablespaces** parameter specifies the INCLUDE XML TABLESPACES option for generated CHECK DATA utility statements.

>**Values:**

>>**ALL** The INCLUDE XML TABLESPACES option is added.

>>**blank** The INCLUDE XML TABLESPACES option is not added; DB2 default utility options are used.

>**Default:**
>>blank

util_check_retry

The **util_check_retry** parameter specifies the RETRY option for generated CHECK DATA utility statements.

Values:

A valid RETRY value for CHECK DATA
Specify a RETRY value in the range 0 - 255. The RETRY option is added with the specified value.

blank The option is not added to the utility statement; DB2 default utility options are used.

Default:
blank

util_check_retry_delay

The **util_check_retry_delay** parameter specifies the RETRY_DELAY option for generated CHECK DATA utility statements.

Values:

A valid RETRY_DELAY value for CHECK DATA
Specify a RETRY_DELAY setting in the range 1 - 1800. The RETRY_DELAY option is added with the specified value.

blank The option is not added to the utility statement; DB2 default utility options are used.

Default:
blank

util_check_scope

The **util_check_scope** parameter specifies the SCOPE option for generated CHECK DATA utility statements.

Values:

P SCOPE PENDING is added.

X SCOPE AUXONLY is added.

A SCOPE ALL is added.

R SCOPE REFONLY is added.

M SCOPE XMLSCHEMAONLY is added. This setting applies only to DB2 V10 and later.

blank The SCOPE option is not added; DB2 default utility options are used.

Default:
blank

util_check_sortdevt

The **util_check_sortdevt** parameter specifies the SORTDEVT option for generated CHECK DATA utility statements.

Values:

A valid SORTDEVT value for CHECK DATA
The SORTDEVT option is added with the specified value, for example, SORTDEVT device-type

Default:
space_unit_name

util_check_sortnum

The **util_check_sortnum** parameter specifies the SORTNUM option for generated CHECK DATA utility statements.

Values:

A valid SORTNUM value for CHECK DATA.
Specify a SORTNUM value in the range 1 - 255. The SORTNUM option is added with the specified value.

Default:
4

util_check_xmlschema

The **util_check_xmlschema** parameter specifies the XMLSCHEMA attribute of the INCLUDE XML TABLESPACES option for generated CHECK DATA utility statements.

Values:

YES The XMLSCHEMA option is added if the INCLUDE XML TABLESPACES option is also added.

NO The XMLSCHEMA option is not added.

Default:
NO

util_clone_template_copyddn1_name

The **util_clone_template_copyddn1_name** parameter specifies the user-provided template name for the first file of COPYDDN.

Values:
Specify a 1- to 8-character DB2 template name.

Default:
CLNCOPY1

util_clone_template_copyddn1_use

The **util_clone_template_copyddn1_use** parameter specifies whether to use a user-provided template for the first COPYDDN file. If a non-blank value is specified, the template name is determined from the **util_clone_template_copyddn1_name** parameter. This parameter is in effect only if the **generate_templates** parameter is set to Y.

Values:
a non-blank value

a non-blank value
A non-blank value indicates that the template name is used if the **generate_templates** is set to Y, and the template exists in the ADBTEMPL file.

Default:
S

util_clone_template_copyddn2_name

The **util_clone_template_copyddn2_name** parameter specifies the user-provided template name for the second file of COPYDDN.

Values:
a DB2 template name; 1 to 8 characters

Default:
CLNCOPY2

util_clone_template_copyddn2_use

The **util_clone_template_copyddn2_use** parameter specifies whether to use a user-provided template for the second COPYDDN file. If a non-blank value is specified, the template name is determined from the **util_clone_template_copyddn2_name** parameter. This parameter is in effect only if the **generate_templates** parameter is set to Y.

Values:

a non-blank value

a non-blank value

A non-blank value indicates that the template name is used if the **generate_templates** parameter is set to Y and the template exists in the ADBTEMPL file.

Default:

S

util_clone_template_discarddn_name

The **util_clone_template_discarddn_name** parameter specifies the user-provided template name for the DISCARDDN file.

Values:

a DB2 template name; 1 to 8 characters

Default:

CLNDISC

util_clone_template_discarddn_use

The **util_clone_template_discarddn_use** parameter specifies whether to use a user-provided template for the DISCARDDN file. If a non-blank value is specified, the template name is determined from the **util_clone_template_discarddn_name** parameter. This parameter is in effect only if the **generate_templates** parameter is set to Y.

Values:

a non-blank value

a non-blank value

A non-blank value indicates that the template name is used if the **generate_templates** parameter is set to Y and the template exists in the ADBTEMPL file.

Default:

S

util_clone_template_errddn_name

The **util_clone_template_errddn_name** parameter specifies the user-provided template name for the ERRDDN file.

Values:

a DB2 template name; 1 to 8 characters

Default:

CLNERR

util_clone_template_errddn_use

The **util_clone_template_errddn_use** parameter specifies whether to use a user-provided template for the ERRDDN file. If a non-blank value is specified, the template name is determined from the **util_clone_template_errddn_name** parameter. This parameter is in effect only if the **generate_templates** parameter is set to Y.

Values:
> a non-blank value

> **a non-blank value**
>> A non-blank value indicates that the template name is used if the **generate_templates** parameter is set to Y and the template exists in the ADBTEMPL file.

Default:
> S

util_clone_template_fccopyddn_name
> The **util_clone_template_fccopyddn_name** parameter specifies the user-provided template name for the FCCOPYDDN file.

> **Values:**
>> a DB2 template name; 1 to 8 characters

> **Default:**
>> CLNFCOPY

util_clone_template_fccopyddn_use parameter
> The **util_clone_template_fccopyddn_use** parameter specifies whether to use a user-provided template for the FCCOPYDDN file. If a non-blank value is specified, the template name is determined from the **util_template_fccopyddn_name** parameter. This parameter is in effect only if the **generate_templates** parameter is set to Y.

> **Values:**
>> a non-blank value

> **Default:**
>> S

util_clone_template_lobcol_name
> The **util_clone_template_lobcol_name** parameter specifies the user provided template name for LOB columns.

> **Values:**
>> a DB2 template name; 1 to 8 characters

> **Default:**
>> CLNLOBC

> The data set name cannot exceed 35 bytes and must be in PDS format. Do not specify a member name

util_clone_template_lobcol_use
> The **util_clone_template_lobcol_use** parameter specifies whether to use a user-provided template for templates related to LOB columns. If a non-blank value is specified, the template name for LOB columns is determined from the **util_clone_template_lobcol_name** parameter. This parameter is in effect only if the **generate_templates** is set to Y.

> **Values:**
>> a non-blank value

> **Default:**
>> S

> A non-blank value indicates that the template name is used if the **generate_templates** is set to Y, and the template exists in the ADBTEMPL file.

util_clone_template_mapddn_name

The **util_clone_template_mapddn_name** parameter specifies the user-provided template name for MAPDDN.

Values:

> **A DB2 template name.**
> Specify a 1- to 8-character DB2 template name.

Default:
CLNMAP

util_clone_template_mapddn_use

The **util_clone_template_mapddn_use** parameter specifies whether to use a user-provided template for the MAPDDN file. If a non-blank value is specified, the template name is determined from the **util_clone_template_mapddn_name** parameter. This parameter is only in effect if the **generate_templates** is set to Y.

Values:

> **A non-blank value.**
> Specify a non-blank value.
>
> A non-blank value indicates that the template name is used if the **generate_templates** is set to Y, and the template exists in the ADBTEMPL file.

Default:
S

util_clone_template_punchddn_name

The **util_clone_template_punchddn_name** parameter specifies the user provided template name for the PUNCHDDN file of the REORG utility.

Values:
a DB2 template name; 1 to 8 characters

Default:
CPUNCH

util_clone_template_punchddn_use

The **util_clone_template_punchddn_use** specifies whether to use a user provided template for the PUNCHDDN file of the REORG utility. If a non-blank value is specified, the template name is determined from the **util_clone_template_punchddn_name** parameter. This parameter is in effect only if the **generate_templates** parameter is set to Y.

Values:
a non-blank value

> **a non-blank value**
> A non-blank value indicates that the template name is used if the **generate_templates** parameter is set to Y and the template exists in the ADBTEMPL file.

Default:
S

util_clone_template_recoveryddn1_name

The **util_clone_template_recoveryddn1_name** parameter specifies the user-provided template name for the first name for RECOVERYDDN.

Values:
>   a DB2 template name; 1 to 8 characters

Default:
>   CLNRCVR1

util_clone_template_recoveryddn1_use
>   The **util_clone_template_recoveryddn1_use** parameter specifies whether to use a user-provided template for the first RECOVERYDDN file. If a non-blank value is specified, the template name is determined from the **util_clone_template_recoveryddn1_name** parameter. This parameter is in effect only if the **generate_templates** parameter is set to Y.

>   **Values:**
>   >   a non-blank value

>   >   **a non-blank value**
>   >   >   A non-blank value indicates that the template name is used if the **generate_templates** parameter is set toY and the template exists in the ADBTEMPL file.

>   **Default:**
>   >   S

util_clone_template_recoveryddn2_name
>   The **util_clone_template_recoveryddn2_name** parameter specifies the user-provided template name for the second name for RECOVERYDDN.

>   **Values:**
>   >   a DB2 template name; 1 to 8 characters

>   **Default:**
>   >   CLNRCVR2

util_clone_template_recoveryddn2_use
>   The **util_clone_template_recoveryddn2_use** parameter specifies whether to use a user-provided template for the second RECOVERYDDN file. If a non-blank value is specified, the template name is determined from the **util_clone_template_recoveryddn2_name** parameter. This parameter is in effect only if **generate_templates** parameter is set to Y.

>   **Values:**
>   >   a non-blank value

>   >   **a non-blank value**
>   >   >   A non-blank value indicates that the template name is used if the **generate_templates** parameter is set to Y and the template exists in the ADBTEMPL file.

>   **Default:**
>   >   S

util_clone_template_unlddn_name
>   The **util_clone_template_unlddn_name** parameter specifies the user provided template name for the UNLDDN file of the REORG utility.

>   **Values:**
>   >   a DB2 template name; 1 to 8 characters

>   **Default:**
>   >   CUNL

util_clone_template_unlddn_use
>   The **util_clone_template_unlddn_use** specifies whether to use a user

provided template for the UNLDDN file of the REORG utility. If a
non-blank value is specified, the template name is determined from the
util_clone_template_unlddn_name parameter. This parameter is in effect
only if the **generate_templates** parameter is set to Y.

Values:

a non-blank value

a non-blank value

A non-blank value indicates that the template name is used
if the **generate_templates** parameter is set to Y and the
template exists in the ADBTEMPL file.

Default:

S

util_clone_template_unload_punchddn_name

The **util_clone_template_unload_punchddn_name** parameter specifies the
user provided template name for the PUNCHDDN file of the UNLOAD
utility.

Values:

a DB2 template name; 1 to 8 characters

Default:

CUPUNCH

util_clone_template_unload_punchddn_use

The **util_clone_template_unload_punchddn_use** specifies whether to use a
user provided template for the PUNCHDDN file of the UNLOAD utility. If
a non-blank value is specified, the template name is determined from the
util_clone_template_unload_punchddn_name parameter. This parameter is
in effect only if the **generate_templates** parameter is set to Y.

Values:

a non-blank value

a non-blank value

A non-blank value indicates that the template name is used
if the **generate_templates** parameter is set to Y and the
template exists in the ADBTEMPL file.

Default:

S

util_clone_template_unload_punchddnc_name

The **util_clone_template_unload_punchddnc_name** parameter specifies the
user provided template name for the DB2 Admin converted version of the
PUNCHDDN file of the UNLOAD utility. Some types of changes require
that the unloaded data be converted by DB2 Admin before the data is
loaded. This parameter controls the user provided template for the
converted load control card for the unloaded data.

Values:

a DB2 template name; 1 to 8 characters

Default:

CUPUNCHC

util_clone_template_unload_punchddnc_use

The **util_clone_template_unload_punchddnc_use** specifies whether to use a
user provided template for the DB2 Admin converted version of the
PUNCHDDN file of the UNLOAD utility. If a non-blank value is specified,

the template name is determined from the **util_clone_template_unload_punchddnc_name** parameter. This parameter is in effect only if the **generate_templates** parameter is set to Y. Some types of changes require that the unloaded data be converted by DB2 Admin before the data is loaded. This parameter controls the user provided template for the converted load control card for the unloaded data.

Values:
a non-blank value

a non-blank value
A non-blank value indicates that the template name is used if the **generate_templates** parameter is set to Y and the template exists in the ADBTEMPL file.

Default:
S

util_clone_template_unload_unlddn_name
The **util_clone_template_unload_unlddn_name** parameter specifies the user provided template name for the UNLDDN file of the UNLOAD utility.

Values:
a DB2 template name; 1 to 8 characters

Default:
CUUNL

util_clone_template_unload_unlddn_use
The **util_clone_template_unload_unlddn_use** specifies whether to use a user provided template for the UNLDDN file of the UNLOAD utility. If a non-blank value is specified, the template name is determined from the **util_clone_template_unload_unlddn_name** parameter. This parameter is in effect only if the **generate_templates** parameter is set to Y.

Values:
a non-blank value

a non-blank value
A non-blank value indicates that the template name is used if the **generate_templates** parameter is set to Y and the template exists in the ADBTEMPL file.

Default:
S

util_clone_template_unload_unlddnc_name
The **util_clone_template_unload_unlddnc_name** parameter specifies the user provided template name for the DB2 Admin converted version of the UNLDDN file of the UNLOAD utility. Some types of changes require that the unloaded data to be converted by DB2 Admin before the data can be loaded. This parameter controls the user provided template for the converted data set for the unloaded data.

Values:
a DB2 template name; 1 to 8 characters

Default:
CUUNLC

util_clone_template_unload_unlddnc_use
The **util_clone_template_unload_unlddnc_use** specifies whether to use a user provided template for the DB2 Admin converted version of the

UNLDDN file of the UNLOAD utility. If a non-blank value is specified, the template name is determined from the **util_clone_template_unload_unlddnc_name** parameter. This parameter is in effect only if the **generate_templates** parameter is set to Y. Some types of changes requires the unloaded data to be converted by DB2 Admin before it can be loaded. This parameter controls the user provided template for the converted data set for the unloaded data.

> **Values:**
>> a non-blank value
>>
>> **a non-blank value**
>>> A non-blank value indicates that the template name is used if the **generate_templates** parameter is set to Y and the template exists in the ADBTEMPL file.
>
> **Default:**
>> S

util_clone_template_workddn1_name
> The **util_clone_template_workddn1_name** parameter specifies the user-provided template name for the first name for WORKDDN.
>
> **Values:**
>> a DB2 template name; 1 to 8 characters
>
> **Default:**
>> CLNWORK1

util_clone_template_workddn1_use
> The **util_clone_template_workddn1_use** parameter specifies whether to use a user-provided template for the first WORKDDN file. If a non-blank value is specified, the template name is determined from the **util_clone_template_workddn1_name** parameter. This parameter is in effect only if the **generate_templates** parameter is set to Y.
>
> **Values:**
>> a non-blank value
>>
>> **a non-blank value**
>>> A non-blank value indicates that the template name is used if the**generate_templates** parameter is set to Y and the template exists in the ADBTEMPL file.
>
> **Default:**
>> S

util_clone_template_workddn2_name
> The **util_clone_template_workddn2_name** parameter specifies the user-provided template name for the second name for WORKDDN.
>
> **Values:**
>> a DB2 template name; 1 to 8 characters
>
> **Default:**
>> CLNWORK2

util_clone_template_workddn2_use
> The **util_clone_template_workddn2_use** parameter specifies whether to use a user-provided template for the second WORKDDN file. If a non-blank value is specified, the template name is determined from the **util_clone_template_workddn2_name** parameter. This parameter is in effect only if the **generate_templates** parameter is set to Y.

Values:

a non-blank value

a non-blank value

A non-blank value indicates that the template name is used if the **generate_templates** parameter is set to Y and the template exists in the ADBTEMPL file.

Default:

S

util_clone_template_xmlcol_name

The **util_clone_template_xmlcol_name** parameter specifies the user-provided template name for XML columns.

Values

a DB2 template name; 1 to 8 characters

Default:

CLNXMLC

The data set name cannot exceed 35 bytes and must be in PDS format. Do not specify a member name.

util_clone_template_xmlcol_use parameter

The **util_clone_template_xmlcol_use** parameter specifies whether to use a user-provided template for templates related to XML columns. If a non-blank value is specified, the template name for XML columns is determined from the **util_clone_template_xmlcol_name** parameter. This parameter is only in effect if the **generate_templates** is set to Y.

Values:

a non-blank value

Default:

S

A non-blank value indicates that the template name is used if the **generate_templates** is set to Y, and the template exists in the ADBTEMPL file.

util_copy_changelimit

The **util_copy_changelimit** parameter specifies the CHANGELIMIT option for generated COPY utility statements.

Values:

Y   The CHANGELIMIT option is added with the user-specified percent_value1 and percent_value2 values.

A   The CHANGELIMIT(ANY) option is added.

blank   The CHANGELIMIT option is not added; DB2 default utility options are used.

Default:

blank

util_copy_changelimit_percent_value1

The **util_copy_changelimit_percent_value1** parameter specifies the CHANGELIMIT percent_value1 option for generated COPY utility statements.

Values:

A percent value allowed by DB2; 0 to 100, 0.0 to 100.0, blank

> **A percent value allowed by DB2; 0 to 100, 0.0 to 100.0**
> The percent_value1 value is specified with the CHANGELIMIT option.

> **blank** The percent_value1 is not specified with the CHANGELIMIT option; DB2 default utility options are used.

Default:
blank

util_copy_changelimit_percent_value2
The **util_copy_changelimit_percent_value2** parameter specifies the CHANGELIMIT percent_value2 option for generated COPY utility statements.

Values:
A percent value allowed by DB2; 0 to 100, 0.0 to 100.0, blank

> **A percent value allowed by DB2; 0 to 100, 0.0 to 100.0**
> The percent_value2 value is specified with the CHANGELIMIT option.

> **blank** The percent_value2is not specified with the CHANGELIMIT option; DB2 default utility options are used.

Default:
blank

util_copy_changelimit_reportonly
The **util_copy_changelimit_reportonly** parameter specifies the CHANGELIMIT REPORTONLY option for generated COPY utility statements.

Values:

> **Y** The REPORTONLY option is added.

> **N** The REPORTONLY option is not added.

Default:
N

util_copy_checkpage
The **util_copy_checkpage** parameter specifies the CHECKPAGE option for generated COPY utility statements.

Values:

> **Y** The CHECKPAGE option is added.

> **N** The CHECKPAGE option is not added.

Default:
N

util_copy_concurrent
The **util_copy_concurrent** parameter specifies the CONCURRENT option for generated COPY utility statements.

Values:

> **Y** The CONCURRENT option is added.

> **N** The CONCURRENT option is not added.

Default:
>N

util_copy_flashcopy
>The **util_copy_flashcopy** parameter specifies the FLASHCOPY option for generated COPY utility statements.
>
>**Values:**
>
>>| | |
>>|---|---|
>>| **Y** | The FLASHCOPY YES option is added. |
>>| **N** | The FLASHCOPY NO option is not added. |
>>| **C** | The FLASHCOPY CONSISTENT option is added. |
>>| **blank** | The FLASHCOPY option is not added; DB2 default utility options are used. |
>
>**Default:**
>>blank

util_copy_full
>The **util_copy_full** parameter specifies the FULL option for generated COPY utility statements.
>
>**Values:**
>
>>| | |
>>|---|---|
>>| **Y** | The FULL YES option is added. |
>>| **N** | The FULL NO option is added. |
>>| **blank** | The FULL option is not added; DB2 default utility options are used. |
>
>**Default:**
>>blank

util_copy_parallel
>The **util_copy_parallel** parameter specifies the PARALLEL option for generated COPY utility statements.
>
>**Values:**
>
>>| | |
>>|---|---|
>>| **0 to 99999** | The PARALLEL option is added as PARALLEL **util_copy_parallel**. Where **util_copy_parallel** is the value specified for this parameter. |
>>| **blank** | The PARALLEL option is not added; DB2 default utility options are used. |
>
>**Default:**
>>blank

util_copy_parallel_tapeunits
>The **util_copy_parallel_tapeunits** parameter specifies the PARALLEL TAPEUNITS option for generated COPY utility statements.
>
>**Values:**
>
>>| | |
>>|---|---|
>>| **0 to 32767** | If the PARALLEL option is added, the TAPEUNITS n option is added. Where n is the value of this parameter. |
>>| **blank** | The TAPEUNITS option is not added; DB2 default utility options are used. |

Default:
　　　blank

util_copy_shrlevel
　　　The **util_copy_shrlevel** parameter specifies the SHRLEVEL option for generated COPY utility statements.

　　　Values:

　　　　　C　　The SHRLEVEL CHANGE option is added.

　　　　　R　　The SHRLEVEL REFERENCE option is added.

　　　　　blank　The SHRLEVEL option is not added; DB2 default utility options are used.

　　　Default:
　　　　　blank

util_load_discards
　　　The **util_load_discards** parameter specifies the DISCARD option for generated LOAD utility statements.

　　　Values:

　　　　　A valid number in the range 0-2147483647

　　　Default:
　　　　　0

util_load_enforce
　　　The **util_load_enforce** parameter specifies the ENFORCE option for generated LOAD utility statements.

　　　Values:

　　　　　YES　The ENFORCE CONSTRAINTS option will be added.

　　　　　NO　The ENFORCE NO option will be added.

　　　Default:
　　　　　YES

util_load_flashcopy
　　　The **util_load_flashcopy** parameter specifies the FLASHCOPY option for generated LOAD utility statements.

　　　Values:

　　　　　Y　　The FLASHCOPY YES option will be added.

　　　　　N　　The FLASHCOPY NO option will be added.

　　　　　C　　The FLASHCOPY CONSISTENT option will be added.

　　　　　blank　The FLASHCOPY option will not be added.

　　　Default:
　　　　　blank

util_load_keepdictionary
　　　The **util_load_keepdictionary** parameter specifies the KEEPDICTIONARY option for generated LOAD utility statements.

　　　Values:

　　　　　YES　The KEEPDICTIONARY option will be added.

　　　　　NO　The KEEPDICTIONARY option will not be added.

Default:
> NO

util_load_log
> The **util_load_log** parameter specifies the LOG option for generated LOAD utility statements.
>
> **Values:**
>> **YES** LOG YES is added.
>>
>> **NO** LOG NO is added.
>>
>> **NOC** LOG NO NOCOPYPEND is added.
>>
>> **blank** The LOG option is not added; DB2 default utility options are used.
>
> **Default:**
>> blank

util_load_parallel
> The **util_load_parallel** parameter specifies the maximum number of subtasks that are to be used in parallel when loading a table space.
>
> **Values:**
>> **YES** The PARALLEL option is added.
>>
>> **integer**
>>> 0-32767. The PARALLEL option is added to the utility statement with the specified value.
>>
>> **blank** The PARALLEL option is not added; DB2 default utility options are used.
>
> **Default:**
>> blank

util_load_resume
> The **util_load_reuse** parameter specifies the RESUME option for generated LOAD utility statements.
>
> **Values:**
>> **YES** RESUME YES is added.
>>
>> **NO** RESUME NO is added.
>>
>> **blank** The RESUME option is not added; DB2 default utility options are used.
>
> **Default:**
>> blank

util_load_replace
> The **util_load_replace** parameter specifies the REPLACE option for generated LOAD utility statements.
>
> **Values:**
>> **YES** The REPLACE option is added.
>>
>> **NO** The REPLACE option is not added.
>>
>> **blank** The REPLACE option is not added; DB2 default utility options are used.

Default:
blank

util_load_reuse
The **util_load_reuse** parameter specifies the REUSE option for generated LOAD utility statements.

Values:

YES The REUSE option will be added.

NO The REUSE option will not be added.

Default:
NO

util_load_shrlevel
The **util_load_shrlevel** parameter specifies the SHRLEVEL option for generated LOAD utility statements.

Values:

N The SHRLEVEL NONE option will be added.

C The SHRLEVEL CHANGE option will be added.

blank The SHRLEVEL option will not be added; DB2 default utility options are used.

Default:
blank

util_load_sortdevt
The **util_load_sortdevt** parameter specifies the SORTDEVT option for generated LOAD utility statements.

Values:

A valid SORTDEVT value for LOAD
The SORTDEVT option will be added with the specified value. For example: SORTDEVT device type.

space_unit_name

Default:
space_unit_name

util_load_sortkeys
The **util_load_sortkeys** parameter specifies the SORTKEYS option for generated LOAD utility statements.

Values:

A valid SORTKEYS value for LOAD. Valid values are 1 through 2147483647.
The SORTKEYS option will be added with the specified value.

0 The SORTKEYS option will not be added.

Default:
0

util_load_sortnum
The **util_load_sortnum** parameter specifies the SORTNUM option for generated LOAD utility statements.

Values:

A valid SORTNUM value for LOAD. Valid values are 1 through
2147483647.
The SORTNUM option will be added with the specified
value.

8

Default:
8

util_reorg_aux

The **util_reorg_aux** parameter specifies the auxiliary option for generated
REORG utility statements. This parameter only applies to DB2 V10 or later.

Values:

YES AUX YES is added.

NO AUX NO is added.

blank The AUX option is not added; DB2 default utility options
are used.

Default:
blank

util_reorg_deadline

The **util_reorg_deadline** parameter specifies the DEADLINE option for
generated REORG utility statements.

Values:

N DEADLINE NONE is added.

timestamp
DEADLINE timestamp is added.

labeled-duration-expression
DEADLINE labeled-duration-expression is added.

blank The DEADLINE option is not added; DB2 default utility
options are used.

Default:
blank

util_reorg_delay

The **util_reorg_delay** parameter specifies the DELAY option for generated
REORG utility statements.

Values:
integer, blank

integer
The DELAY option is added to the utility statement with
the specified value. *Integer* is the number of seconds.

blank The DELAY option is not added to the utility statement;
DB2 default utility options are used.

Default:
blank

util_reorg_drain

The **util_reorg_drain** parameter specifies the DRAIN option for generated REORG utility statements.

Values:

> **W** The DRAIN WRITERS option is added to the utility statement.
>
> **A** The DRAIN ALL option is added to the utility statement.
>
> **blank** The DRAIN ALL option is not added to the utility statement; DB2 default utility options are used.

Default:
blank

util_reorg_drain_wait

The **util_reorg_drain_wait** parameter specifies the DRAIN_WAIT option for generated REORG utility statements.

Values:

> *integer*, blank
>
> *integer*
>> A valid DRAIN_WAIT value for REORG is a value between 0 - 1800. The DRAIN_WAIT option is added with the specified value.
>
> **blank** The DRAIN ALL option is not added to the utility statement; DB2 default utility options are used.

Default:
blank

util_reorg_flashcopy

The **util_reorg_flashcopy** parameter specifies the FLASHCOPY option for generated REORG utility statements.

Values:

> **Y** FLASHCOPY YES is added.
>
> **C** FLASHCOPY CONSISTENT is added.
>
> **N** FLASHCOPY NO is added.
>
> **blank** The FLASHCOPY option is not added; DB2 default utility options are used.

Default:
blank

util_reorg_fastswitch

The **util_reorg_fastswitch** parameter specifies the FASTSWITCH option for generated REORG utility statements.

Values:

> **Y** FASTSWITCH YES is added.
>
> **N** FASTSWITCH NO is added.
>
> **blank** The FASTSWITCH option is not added; DB2 default utility options are used.

Default:
>blank

util_reorg_index_clone
>The **util_reorg_index_clone** parameter specifies the CLONE option for generated REORG INDEX utility statements.

>**Values:**

>>**Y**   Clone is added.

>>**N**   Clone is not is added.

>**Default:**
>>N

util_reorg_index_fastswitch
>The **util_reorg_index_fastswitch** parameter specifies the FASTSWITCH option for generated REORG INDEX utility statements.

>**Values:**

>>**Y**   FASTSWITCH YES is added.

>>**N**   FASTSWITCH NO is added.

>>**blank** The FASTSWITCH option is not added; DB2 default utility options are used.

>**Default:**
>>blank

util_reorg_index_flashcopy
>The **util_reorg_index_flashcopy** parameter specifies the FLASHCOPY option for generated REORG INDEX utility statements.

>**Values:**

>>**Y**   FLASHCOPY YES is added.

>>**C**   FLASHCOPY CONSISTENT is added.

>>**N**   FLASHCOPY NO is added.

>>**blank** The FLASHCOPY option is not added; DB2 default utility options are used.

>**Default:**
>>blank

util_reorg_index_leafdistlimit
>The **util_reorg_index_leafdistlimit** parameter specifies the LEAFDISTLIMIT option for generated REORG INDEX utility statements.

>**Values:**
>>A valid LEAFDISTLIMIT value for REORG INDEX, blank

>>**A valid LEAFDISTLIMIT value for REORG INDEX**
>>>LEAFDISTLIMIT is added with the specified value.

>>**blank** The LEAFDISTLIMIT option is not added; DB2 default utility options are used.

>**Default:**
>>blank

util_reorg_index_preformat

The **util_reorg_index_preformat** parameter specifies the PREFORMAT option for generated REORG INDEX utility statements.

Values:

Y	PREFORMAT is added.
N	PREFORMAT is not added

Default:
N

util_reorg_index_reportonly

The **util_reorg_index_reportonly** parameter specifies the REPORTONLY option for generated REORG INDEX utility statements.

Values:

Y	REPORTONLY is added.
N	REPORTONLY is not added.

Default:
N

util_reorg_index_reuse

The **util_reorg_index_reuse** parameter specifies the REUSE option for generated REORG INDEX utility statements.

Values:

Y	REUSE is added
N	REUSE is not added.

Default:
N

util_reorg_index_shrlevel

The **util_reorg_index_shrlevel** parameter specifies the SHRLEVEL option for generated REORG INDEX utility statements.

Values:

N	SHRLEVEL NONE is added.
C	SHRLEVEL CHANGE is added. However, the option might not be specified, or might be converted to SHRLEVEL REFERENCE for some generated REORG index statements. SHRLEVEL CHANGE is processed based on SHRLEVEL REFERENCE restrictions that are described in the DB2 Utility Reference manual.
R	SHRLEVEL REFERENCE is added. However, the option might not be specified for some generated REORG index statements. SHRLEVEL REFERENCE is processed based on SHRLEVEL REFERENCE restrictions that are described in the DB2 Utility Reference manual.
blank	The SHRLEVEL option is not added; DB2 default utility options are used.

Default:
blank

util_reorg_index_sortdevt

The **util_reorg_index_sortdevt** parameter specifies the SORTDEVT option for generated REORG INDEX utility statements.

Values:

A valid SORTDEVT value for REORG INDEX, blank

A valid SORTDEVT value for REORG INDEX
The SORTDEVT option is added with the specified value, for example the SORTDEVT device-type value.

blank   The SORTDEVT option is not added; DB2 default utility options are used.

Default:

space_unit_name

util_reorg_index_sortnum

The **util_reorg_index_sortnum** parameter specifies the SORTNUM option for generated REORG INDEX utility statements.

Values:

A valid SORTNUM value for REORG INDEX, blank

A valid SORTNUM value for REORG INDEX
The SORTNUM option is added with the specified value.

blank   The SORTNUM option is not added; DB2 default utility options are used.

Default:

4

util_reorg_indreflimit

The **util_reorg_indreflimit** parameter specifies the INDREFLIMIT option for generated REORG utility statements.

Values:

A valid INDREFLIMIT value for REORG, blank

A valid INDREFLIMIT value for REORG
INDREFLIMIT is added with the specified value.

blank   The INDREFLIMIT option is not added; DB2 default utility options are used.

Default:

blank

util_reorg_keepdictionary

The **util_reorg_keepdictionary** parameter specifies the KEEPDICTIONARY option for generated REORG utility statements.

Values:

Y   KEEPDICTIONARY is added.

N   KEEPDICTIONARY is not added.

Default:

N

util_reorg_listparts

The **util_reorg_listparts** parameter specifies the LISTPARTS option for generated REORG utility statements that use listdef. The LISTPARTS

option can be used only when the LIST keyword is specified. The **util_reorg_listparts** and **util_reorg_parallel** parameters are mutually exclusive.

Values:

> **Positive integer**
> > The LISTPARTS option is added with the specified value.
>
> **blank** The LISTPARTS option is not added; DB2 default utility options are used.

Default:
> blank

util_reorg_log
> The **util_reorg_log** parameter specifies the LOG option for generated REORG utility statements.
>
> **Values:**
>
> > **Y** LOG YES is added.
> >
> > **N** LOG NO is added.
> >
> > **blank** The LOG option is not added; DB2 default utility options are used.
>
> **Default:**
> > N

util_reorg_logranges
> The **util_reorg_logranges** parameter specifies the LOGRANGES option for generated REORG utility statements.
>
> **Values:**
>
> > **YES** LOGRANGES YES is added.
> >
> > **NO** LOGRANGES NO is added.
> >
> > **blank** The LOGRANGES option is not added; DB2 default utility options are used.
>
> **Default:**
> > blank

util_reorg_longlog
> The **util_reorg_longlog** parameter specifies the LONGLOG option for generated REORG utility statements.
>
> **Values:**
>
> > **C** The LONGLOG CONTINUE option is added to the utility statement.
> >
> > **T** The LONGLOG TERM option is added to the utility statement.
> >
> > **D** The LONGLOG DRAIN option is added to the utility statement.
> >
> > **blank** The LONGLOG option is not added to the utility statement; DB2 default utility options are used.
>
> **Default:**
> > blank

util_reorg_mappingdatabase

The **`util_reorg_mappingdatabase`** parameter specifies the MAPPINGDATABASE option for generated REORG utility statements.

Values:

A database name; 1 to 8 characters.

Default:

blank

util_reorg_maptable_name

The **`util_reorg_maptable_name`** parameter specifies the MAPTABLE name for generated REORG utility statements.

Values:

Valid table owner name; 1 to 128 characters

Default:

blank

util_reorg_maptable_owner

The **`util_reorg_maptable_owner`** parameter specifies the MAPTABLE owner for generated REORG utility statements.

Values:

Valid table owner name; 1 to 128 characters

Default:

blank

util_reorg_maxro

The **`util_reorg_maxro`** parameter specifies the MAXRO option for generated REORG utility statements.

Values:

integer

The MAXRO option is added to the utility statement with the specified value.

D   The MAXRO DEFER option is added to the utility statement.

blank The MAXRO option is not added; DB2 default utility options are used.

Default:

blank

util_reorg_newmaxro

The **`util_reorg_newmaxro`** parameter specifies the NEWMAXRO option for generated REORG utility statements.

Values:

NONE

NEWMAXRO NONE is added.

integer

NEWMAXRO integer is added.

blank The NEWMAXRO option is not added; DB2 default utility options are used.

Default:

blank

util_reorg_nosysrec

The **util_reorg_nosysrec** parameter specifies the NOSYSREC option for generated REORG utility statements.

Values:

    Y   NOSYSREC is added.

    N   NOSYSREC is not added.

Default:
    N

util_reorg_offposlimit

The **util_reorg_offposlimit** parameter specifies the OFFPOSLIMIT option for generated REORG utility statements.

Values:

    A valid OFFPOSLIMIT value for REORG
        OFFPOSLIMIT is added with the specified value.

    blank The OFFPOSLIMIT option is not added; DB2 default utility options are used.

Default:
    blank

util_reorg_parallel

The **util_reorg_parallel** specifies the maximum number of subtasks that are to be started in parallel to reorganize a table space.

Values:

    YES   The PARALLEL option is added.

    integer
        A valid PARALLEL value for REORG is a value0 - 32767. The PARALLEL option is added to the utility statement with the specified value.

    blank The PARALLEL option is not added; DB2 default utility options are used.

Default:
    blank

util_reorg_preformat

The **util_reorg_preformat** parameter specifies the PREFORMAT option for generated REORG utility statements.

Values:

    Y   PREFORMAT is added.

    N   PREFORMAT is not added.

Default:
    N

util_reorg_recluster

The **util_reorg_recluster** parameter specifies the RECLUSTER option of SORTDATA NO for generated REORG utility statements.

Values:

    YES   RECLUSTER YES is added.

NO RECLUSTER NO is added.

blank The RECLUSTER option is not added; DB2 default utility options are used.

Default:
blank

util_reorg_retry

The **util_reorg_retry** parameter specifies the RETRY option for generated REORG utility statements.

Values:

A valid RETRY value for REORG
The RETRY option is added with the specified value. The value must be an integer 0 - 255.

blank The RETRY option is not added; DB2 default utility options are used.

Default:
blank

util_reorg_retry_delay

The **util_reorg_retry_delay** parameter specifies the RETRY_DELAY option for generated REORG utility statements.

Values:

A valid RETRY_DELAY value for REORG
The RETRY_DELAY option is added with the specified value. The value must be an integer 1 - 1800.

blank The RETRY_DELAY option is not added to the utility statement; DB2 default utility options are used.

Default:
blank

util_reorg_reuse

The **util_reorg_reuse** parameter specifies the REUSE option for generated REORG utility statements.

Values:

Y REUSE is added.

N REUSE is not added.

Default:
N

util_reorg_shrlevel

The **util_reorg_shrlevel** parameter specifies the SHRLEVEL option for generated REORG utility statements.

Values:

N SHRLEVEL NONE is added.

C SHRLEVEL CHANGE is added. However, the option might not be specified, or might be converted to SHRLEVEL REFERENCE for some generated REORG table space statements. SHRLEVEL CHANGE is processed based

> on SHRLEVEL REFERENCE restrictions that are described in the DB2 Utility Reference manual.

> **R** SHRLEVEL REFERENCE is added. However, the option might not be specified for some generated REORG table space statements. SHRLEVEL REFERENCE is processed based on SHRLEVEL REFERENCE restrictions that are described in the DB2 Utility Reference manual.

> **blank** The SHRLEVEL option is not added; DB2 default utility options are used.

Default:
blank

util_reorg_sortdata

The **util_reorg_sortdata** parameter specifies the SORTDATA option for generated REORG utility statements.

Values:

> **Y** SORTDATA is added.

> **N** SORTDATA NO is added.

> **blank** The SORTDATA option is not added; DB2 default utility options are used.

Default:
blank

util_reorg_sortdevt

The **util_reorg_sortdevt** parameter specifies the SORTDEVT option for generated REORG utility statements.

Values:

> **A valid SORTDEVT value for REORG**
> The SORTDEVT option is added with the specified value. E.g. SORTDEVT device-type

> **blank** The SORTDEVT option is not added; DB2 default utility options are used.

Default:
space_unit_name

util_reorg_sortkeys

The **util_reorg_sortkeys** parameter specifies the SORTKEYS option for generated REORG utility statements.

Values:

> **Y** SORTKEYS is added.

> **N** SORTKEYS is not added.

Default:
N

util_reorg_sortnum

The **util_reorg_sortnum** parameter specifies the SORTNUM option for generated REORG utility statements.

Values:

A valid SORTNUM value for REORG
　　The SORTNUM option is added with the specified value.

blank　The SORTNUM option is not added; DB2 default utility options are used.

Default:
　4

util_reorg_statistics

The **util_reorg_statistics** specifies the STATISTICS option for generated REORG utility statements.

Values:

Y　　The STATISTICS option is added.

N　　The STATISTICS option is not added. Any other specified REORG statistics options are not used.

blank　The STATISTICS option is conditionally added. It is added if a REORG statistics option was explicitly specified. For example, if a value for SAMPLE was specified using the **util_reorg_statistics_table_sample** parameter.

Default:
　blank

util_reorg_statistics_forcerollup

The **util_reorg_statistics_forcerollup** parameter specifies the FORCEROLLUP option for generated REORG utility statements.

Values:

Y　　FORCEROLLUP YES is added.

N　　FORCEROLLUP NO is added.

blank　The FORCEROLLUP option is not added; DB2 default utility options are used.

Default:
　blank

util_reorg_statistics_history

The **util_reorg_statistics_history** parameter specifies the HISTORY option for generated REORG utility statements.

Values:

A　　HISTORY ALL is added.

P　　HISTORY ACCESSPATH is added.

S　　HISTORY SPACE is added.

N　　HISTORY NONE is added.

blank　The HISTORY option is not added; DB2 default utility options are used.

Default:
　blank

util_reorg_statistics_index_histogram

The **util_reorg_statistics_index_histogram** parameter specifies the HISTOGRAM option for generated REORG utility statements.

Values:

> **Y** The HISTOGRAM option is added.
>
> **N** The HISTOGRAM option is not added. Any other specified HISTOGRAM options are not used.
>
> **blank** The HISTOGRAM option is conditionally added. It is added if a value is specified for the **util_reorg_statistics_index_numcols** parameter or the **util_reorg_statistics_index_numquantiles** parameter.

Default:
> blank

util_reorg_statistics_index_numcols

> The **util_reorg_statistics_index_numcols** specifies the NUMCOLS option for generated REORG utility statements. If a value is not specified for the **util_reorg_statistics_index_numcols** parameter but a value is specified for the **util_reorg_statistics_index_numquantiles** parameter, then NUMCOLS 1 is added to generated REORG statements.

Values:

> **1 - 64** The NUMCOLS option is added with the specified value.
>
> **blank** The NUMCOLS option is not added; DB2 default utility options are used.

Default:
> blank

util_reorg_statistics_index_numquantiles

> The **util_reorg_statistics_index_numquantiles** specifies the NUMQUANTILES option for generated REORG utility statements.

Values:

> **1 - 100** The NUMQUANTILES option is added with the specified value.
>
> **blank** The NUMQUANTILES option is not added; DB2 default utility options are used.

Default:
> blank

util_reorg_statistics_report

> The **util_reorg_statistics_report** parameter specifies the REPORT option for generated REORG utility statements.

Values:

> **Y** REPORT YES is added.
>
> **N** REPORT NO is added.
>
> **blank** The REPORT option is not added; DB2 default utility options are used.

Default:
> blank

util_reorg_statistics_table_sample

> The **util_reorg_statistics_table_sample** parameter specifies the SAMPLE option for generated REORG utility statements.

1 to 100
> The SAMPLE option is added with the specified value.

blank The SAMPLE option is not added; DB2 default utility options are used.

Default:
> blank

util_reorg_statistics_update
> The **util_reorg_statistics_update** parameter specifies the UPDATE option for generated REORG utility statements.

> **Values:**

>> **A**   UPDATE ALL is added.

>> **P**   UPDATE ACCESSPATH is added.

>> **S**   UPDATE SPACE is added.

>> **N**   UPDATE NONE is added.

>> **blank** The UPDATE option is not added; DB2 default utility options are used.

> **Default:**
>> blank

util_reorg_switchtime
> The **util_reorg_switchtime** parameter specifies the SWITCHTIME option for generated REORG utility statements.

> **Values:**

>> **NONE**
>>> SWITCHTIME NONE is added.

>> **timestamp**
>>> SWITCHTIME timestamp is added.

>> **labeled-duration-expression**
>>> SWITCHTIME labeled-duration-expression is added.

>> **blank** The SWITCHTIME option is not added; DB2 default utility options are used.

> **Default:**
>> blank

util_reorg_timeout
> The **util_reorg_timeout** parameter specifies the TIMEOUT option for generated REORG utility statements.

> **Values:**

>> **T**   The TIMEOUT TERM option is added to the utility statement.

>> **A**   The TIMEOUT ABEND option is added to the utility statement.

>> **blank** The TIMEOUT option is not added to the utility statement; DB2 default utility options are used.

> **Default:**
>> blank

util_runstats_history

The **util_runstats_history** parameter specifies the HISTORY option for generated RUNSTATS utility statements.

Values:

A   HISTORY ALL is added.

P   HISTORY ACCESSPATH is added.

S   HISTORY SPACE is added.

N   HISTORY NONE is added.

blank   The HISTORY option is not added; DB2 default utility options are used.

Default:

blank

util_runstats_report

The **util_runstats_report** parameter specifies the REPORT option for generated RUNSTATS utility statements.

Values:

Y   REPORT YES is added.

N   REPORT NO is added.

blank   The REPORT option is not added; DB2 default utility options are used.

Default:

blank

util_runstats_shrlevel

The **util_runstats_shrlevel** parameter specifies the SHRLEVEL option for generated RUNSTATS utility statements.

Values:

C   SHRLEVEL CHANGE is added.

R   SHRLEVEL REFERENCE is added.

blank   The SHRLEVEL option is not added; DB2 default utility options are used.

Default:

blank

util_runstats_update

The **util_runstats_update** parameter specifies the UPDATE option for generated RUNSTATS utility statements.

Values:

A   UPDATE ALL is added.

P   UPDATE ACCESSPATH is added.

S   UPDATE SPACE is added.

N   UPDATE NONE is added.

blank   The UPDATE option is not added; DB2 default utility options are used.

Default:
> blank

util_template_copyddn1_name
> The **util_template_copyddn1_name** parameter specifies the user provided template name for the first file of COPYDDN.
>
> **Values:**
>
> > **A DB2 template name**
> > > The DB2 template name can be 1 to 8 character in length.
>
> **Default:**
> > COPY1

util_template_copyddn1_use
> The **util_template_copyddn1_use** parameter specifies whether to use a user provided template for the first COPYDDN file. If a non-blank value is specified, the template name is determined from the **util_template_copyddn1_name** parameter. This parameter is in effect only if the **generate_templates** parameter is set to Y.
>
> **Values:**
>
> > **A non-blank value**
> > > A non-blank value indicates that the template name is used if the **generate_templates** parameter is set to Y and the template exists in the ADBTEMPL file.
>
> **Default:**
> > S

util_template_copyddn2_name
> The **util_template_copyddn2_name** parameter specifies the user provided template name for the second file of COPYDDN.
>
> **Values:**
>
> > **A DB2 template name**
> > > The DB2 template name can be 1 to 8 character in length.
>
> **Default:**
> > COPY2

util_template_copyddn2_use
> The **util_template_copyddn2_use** parameter specifies whether to use a user provided template for the second COPYDDN file. If a non-blank value is specified, the template name is determined from the **util_template_copyddn2_name** parameter. This parameter is in effect only if the **generate_templates** parameter is set to Y.
>
> **Values:**
>
> > **A non-blank value**
> > > A non-blank value indicates that the template name is used if the **generate_templates** parameter is set to Y and the template exists in the ADBTEMPL file.
>
> **Default:**
> > S

util_template_discarddn_name
> The **util_template_discarddn_name** parameter specifies the user provided template name for the DISCARDDN file.

Values:

> **A DB2 template name**
>> The DB2 template name can be 1 to 8 character in length.

Default:
> DISC

util_template_discarddn_use
> The **util_template_discarddn_use** parameter specifies whether to use a user provided template for the DISCARDDN file. If a non-blank value is specified, the template name is determined from the **util_template_discarddn_name** parameter. This parameter is in effect only if the **generate_templates** parameter is set to Y.

> **Values:**

>> **A non-blank value**
>>> A non-blank value indicates that the template name is used if the **generate_templates** parameter is set to Y, and the template exists in the ADBTEMPL file.

> **Default:**
>> S

util_template_errddn_devtype
> The **util_template_errddn_devtype** parameter specifies whether the ERRDDN template is on a tape-like device, or on a DASD device.

> **Values:**

>> **TAPE** A removal media device, such as 3490 tape, or a 3490E tape drive.

>> **DASD**
>>> A magnetic disk storage device, such as a direct access storage device (DASD).

> **Default:**
>> DASD

util_template_errddn_name
> The **util_template_errddn_name** parameter specifies the user provided template name for the ERRDDN file.

> **Values:**

>> **A DB2 template name**
>>> The DB2 template name can be 1 to 8 character in length.

> **Default:**
>> ERR

util_template_errddn_use
> The **util_template_errddn_use** parameter specifies whether to use a user provided template for the ERRDDN file. If a non-blank value is specified, the template name is determined from the **util_template_errddn_name** parameter. This parameter is in effect only if the **generate_templates** parameter is set to Y.

> **Values:**

>> **A non-blank value**
>>> A non-blank value indicates that the template name is used

if the **generate_templates** parameter is set to Y and the template exists in the ADBTEMPL file.

Default:
S

util_template_fccopyddn_name
The **util_template_fccopyddn_name** parameter specifies the user provided template name for the FCCOPYDDN file.

Values:

A DB2 template name
The DB2 template name can be 1 to 8 character in length.

Default:
FCOPY

util_template_fccopyddn_use
The **util_template_fccopyddn_use** parameter specifies whether to use a user provided template for the FCCOPYDDN file. If a non-blank value is specified, the template name is determined from the **util_template_fccopyddn_name** parameter. This parameter is in effect only if the **generate_templates** parameter is set to Y.

Values:

A non-blank value
A non-blank value indicates that the template name is used if the **generate_templates** parameter is set to Y and the template exists in the ADBTEMPL file.

Default:
S

util_template_lobcol_name
The **util_template_lobcol_name** parameter specifies the user provided template name for LOB columns.

Values:

A DB2 template name
The DB2 template name can be 1 to 8 character in length.

Default:
LOBC

util_template_lobcol_use
The **util_template_lobcol_use** parameter specifies whether to use a user provided template for templates related to LOB columns. If a non-blank value is specified, the template name for LOB columns is determined from the **util_template_lobcol_name** parameter. This parameter is in effect only if the **generate_templates** parameter is set to Y.

Values:

A non-blank value
A non-blank value indicates that the template name is used if the **generate_templates** parameter is set to Y and the template exists in the ADBTEMPL file.

Default:
S

util_template_mapddn_devtype

The **util_template_mapddn_devtype** parameter specifies whether the MAPDDN template is on a tape-like device, or on a DASD device.

Values:

TAPE A removal media device, such as 3490 tape, or a 3490E tape drive.

DASD
A magnetic disk storage device, such as a direct access storage device (DASD).

Default:
DASD

util_template_mapddn_name

The **util_template_mapddn_name** parameter specifies the user provided template name for MAPDDN.

Values:

A DB2 template name
The DB2 template name can be 1 to 8 character in length.

Default:
MAP

util_template_mapddn_use

The **util_template_mapddn_use** parameter specifies whether to use a user-provided template for the MAPDDN file. If a non-blank value is specified, the template name is determined from the **util_template_mapddn_name** parameter. This parameter is only in effect if the **generate_templates** parameter is set to Y.

Values:

A non-blank value.
A non-blank value indicates that the template name is used if the **generate_templates** parameter is set to Y, and the template exists in the ADBTEMPL file.

Default:
S

util_template_punchddn_name

The **util_template_punchddn_name** parameter specifies the user provided template name for the PUNCHDDN file of the REORG utility.

Values:

A DB2 template name
The DB2 template name can be 1 to 8 character in length.

Default:
PUNCH

util_template_punchddn_use

The **util_template_punchddn_use** specifies whether to use a user provided template for the PUNCHDDN file of the REORG utility. If a non-blank value is specified, the template name is determined from the **util_template_punchddn_name** parameter. This parameter is in effect only if the **generate_templates** parameter is set to Y.

Values:

A non-blank value

A non-blank value indicates that the template name is used
if the **generate_templates** parameter is set to Y and the
template exists in the ADBTEMPL file.

Default:
S

util_template_recoveryddn1_name
The **util_template_recoveryddn1_name** parameter specifies the
user-provided template name for the first file of RECOVERYDDN.

Values:

A DB2 template name
The DB2 template name can be 1 to 8 character in length.

Default:
RCVR1

util_template_recoveryddn1_use
The **util_template_recoveryddn1_use** parameter specifies whether to use a
user-provided template for the first RECOVERYDDN file. If a non-blank
value is specified, the template name is determined from the
util_template_recoveryddn1_name parameter. This parameter is only in
effect if the **generate_templates** parameter is set to Y.

Values:

A non-blank value.
A non-blank value indicates that the template name is used
if the **generate_templates** parameter is set to Y, and the
template exists in the ADBTEMPL file.

Default:
S

util_template_recoveryddn2_name
The **util_template_recoveryddn2_name** parameter specifies the
user-provided template name for the second file of RECOVERYDDN.

Values:

A DB2 template name
The DB2 template name can be 1 to 8 character in length.

Default:
RCVR2

util_template_recoveryddn2_use
The **util_template_recoveryddn2_use** parameter specifies whether to use a
user-provided template for the second RECOVERYDDN file. If a non-blank
value is specified, the template name is determined from the
util_template_recoveryddn2_name parameter. This parameter is only in
effect if the **generate_templates** parameter is set to Y.

Values:

A non-blank value.
A non-blank value indicates that the template name is used
if the **generate_templates** parameter is set to Y, and the
template exists in the ADBTEMPL file.

Default:
　　S

util_template_unlddn_name
　　The **util_template_unlddn_name** parameter specifies the user provided template name for the UNLDDN file of the REORG utility.

　　Values:

　　　　A DB2 template name
　　　　　　The DB2 template name can be 1 to 8 character in length.

　　Default:
　　　　UNL

util_template_unlddn_use
　　The **util_template_unlddn_use** parameter specifies whether to use a user provided template for the UNLDDN file of the REORG utility. If a non-blank value is specified, the template name is determined from the **util_template_unlddn_name** parameter. This parameter is in effect only if the **generate_templates** parameter is set to Y.

　　Values:

　　　　A non-blank value
　　　　　　A non-blank value indicates that the template name is used if the **generate_templates** parameter is set to Y and the template exists in the ADBTEMPL file.

　　Default:
　　　　S

util_template_unload_punchddn_name
　　The **util_template_unload_punchddn_name** parameter specifies the user provided template name for the PUNCHDDN file of the UNLOAD utility.

　　Values:

　　　　A DB2 template name
　　　　　　The DB2 template name can be 1 to 8 character in length.

　　Default:
　　　　UPUNCH

util_template_unload_punchddn_use
　　The **util_template_unload_punchddn_use** specifies whether to use a user provided template for the PUNCHDDN file of the UNLOAD utility. If a non-blank value is specified, the template name is determined from the **util_template_unload_punchddn_name** parameter. This parameter is in effect only if the **generate_templates** parameter is set to Y.

　　Values:

　　　　A non-blank value
　　　　　　A non-blank value indicates that the template name is used if the **generate_templates** parameter is set to Y and the template exists in the ADBTEMPL file.

　　Default:
　　　　S

util_template_unload_punchddnc_name
　　The **util_template_unload_punchddnc_name** parameter specifies the user provided template name for the DB2 Admin converted version of the

PUNCHDDN file of the UNLOAD utility. Some types of changes require that the unloaded data to be converted by DB2 Admin before the data is loaded. This parameter controls the user provided template for the converted load control card for the unloaded data.

Values:

A DB2 template name
The DB2 template name can be 1 to 8 character in length.

Default:
UPUNCHC

util_template_unload_punchddnc_use
The **util_template_unload_punchddnc_use** specifies whether to use a user provided template for the PUNCHDDN file of the UNLOAD utility. If a non-blank value is specified, the template name is determined from the **util_template_unload_punchddnc_name** parameter. This parameter is in effect only if the **generate_templates** parameter is set to Y. Some types of changes requires the unloaded data to be converted by DB2 Admin before it can be loaded. This parameter controls the user provided template for the converted load control card for the unloaded data.

Values:

A non-blank value
A non-blank value indicates that the template name is used if the **generate_templates** parameter is set to Y and the template exists in the ADBTEMPL file.

Default:
S

util_template_unload_unlddn_devtype
The **util_template_unload_unlddn_devtype** specifies whether **util_template_unload_unlddn_name** is on removable media or on a DASD device.

Values:

TAPE A removal media device, such as 3490 tape, or a 3490E tape drive.

DASD
A magnetic disk storage device, such as a direct access storage device (DASD).

blank The DEVTYPE option is not added; DB2 default utility options are used.

Default:
blank

util_template_unload_unlddn_name
The **util_template_unload_unlddn_name** parameter the user provided template name for the UNLDDN file of the UNLOAD utility.

Values:

A DB2 template name
The DB2 template name can be 1 to 8 character in length.

Default:
UUNL

util_template_unload_unlddn_use

The **util_template_unload_unlddn_use** specifies whether to use a user provided template for the UNLDDN file of the UNLOAD utility. If a non-blank value is specified, the template name is determined from the **util_template_unload_unlddn_name** parameter. This parameter is in effect only if the **generate_templates** parameter is set to Y.

Values:

> **A non-blank value**
> A non-blank value indicates that the template name is used if the **generate_templates** parameter is set to Y and the template exists in the ADBTEMPL file.

Default:
S

util_template_unload_unlddnc_name

The **util_template_unload_unlddnc_name** parameter specifies the user provided template name for the DB2 Admin converted version of the UNLDDN file of the UNLOAD utility. Some types of changes require that the unloaded data to be converted by DB2 Admin before the data is loaded. This parameter controls the user provided template for the converted data set for the unloaded data.

Values:

> **A DB2 template name**
> The DB2 template name can be 1 to 8 character in length.

Default:
UUNLC

util_template_unload_unlddnc_use

The **util_template_unload_unlddnc_use** specifies whether to use a user provided template for the DB2 Admin converted version of the UNLDDN file of the UNLOAD utility. If a non-blank value is specified, the template name is determined from the **util_template_unload_unlddnc_name** parameter. This parameter is in effect only if the **generate_templates** parameter is set to Y. Some types of changes require that the unloaded data to be converted by DB2 Admin before the data is loaded. This parameter controls the user provided template for the converted data set for the unloaded data.

Values:

> **A non-blank value**
> A non-blank value indicates that the template name is used if the **generate_templates** parameter is set to Y and the template exists in the ADBTEMPL file.

Default:
S

util_template_workddn1_devtype

The **util_template_workddn1_devtype** parameter specifies whether the WORKDDN1 template is on a tape-like device, or on a DASD device.

Values:

> **TAPE** A removal media device, such as 3490 tape, or a 3490E tape drive.

DASD
> A magnetic disk storage device, such as a direct access storage device (DASD).

Default:
> DASD

util_template_workddn1_name
> The **util_template_workddn1_name** parameter specifies the user provided template name for the first name for WORKDDN.

Values:

> **A DB2 template name**
> > The DB2 template name can be 1 to 8 character in length.

Default:
> WORK1

util_template_workddn1_use
> The **util_template_workddn1_use** parameter specifies whether to use a user provided template for the first WORKDDN file. If a non-blank value is specified, the template name is determined from the **util_template_workddn1_name** parameter. This parameter is in effect only if the **generate_templates** parameter is set to Y.

Values:

> **A non-blank value**
> > A non-blank value indicates that the template name is used if the **generate_templates** parameter is set to Y and the template exists in the ADBTEMPL file.

Default:
> S

util_template_workddn2_devtype
> The **util_template_workddn2_devtype** parameter specifies whether the WORKDDN2 template is on a tape-like device, or on a DASD device.

Values:

> **TAPE** A removal media device, such as 3490 tape, or a 3490E tape drive.

> **DASD**
> > A magnetic disk storage device, such as a direct access storage device (DASD).

Default:
> DASD

util_template_workddn2_name
> The **util_template_workddn2_name** parameter specifies the user provided template name for the second name for WORKDDN.

Values:

> **A DB2 template name**
> > The DB2 template name can be 1 to 8 character in length.

Default:
> WORK2

util_template_workddn2_use

The **util_template_workddn2_use** parameter specifies whether to use a user provided template for the second WORKDDN file. If a non-blank value is specified, the template name is determined from the **util_template_workddn2_name** parameter. This parameter is in effect only if the **generate_templates** parameter is set to Y.

Values:

A non-blank value
A non-blank value indicates that the template name is used if the **generate_templates** parameter is set to Y and the template exists in the ADBTEMPL file.

Default:
S

util_template_xmlcol_name

The **util_template_xmlcol_name** parameter specifies the user provided template name for XML columns.

Values:

A DB2 template name
The DB2 template name can be 1 to 8 character in length.

Default:
XMLC

util_template_xmlcol_use

The **util_template_xmlcol_use** parameter specifies whether to use a user provided template for templates related to XML columns. If a non-blank value is specified, the template name for XML columns is determined from the **util_template_xmlcol_name** parameter. This parameter is in effect only if the **generate_templates** parameter is set to Y.

Values:

A non-blank value
A non-blank value indicates that the template name is used if the **generate_templates** parameter is set to Y and the template exists in the ADBTEMPL file.

Default:
S

util_unload_dbcs_ccsid

The **util_unload_dbcs_ccsid** parameter specifies the DBCS CCSID option for generated UNLOAD utility statements.

Values:

A valid CCSID value
The CCSID option is added with the specified value. E.g. CCSID(**util_unload_sbcs_ccsid**, **util_unload_mixed_ccsid**, **util_unload_dbcs_ccsid**)

blank The value is omitted from the CCSID option; DB2 default utility options are used.

Default:
blank

util_unload_encodingscheme

The **util_unload_encodingscheme** parameter specifies the ENCODINGSCHEME option for generated UNLOAD utility statements.

Values:

    E   EBCDIC is added.

    A   ASCII is added.

    U   UNICODE is added.

    blank   No encoding scheme option is added; DB2 default utility options are used.

Default:
    blank

util_unload_float

The **util_unload_float** parameter specifies the FLOAT option for generated UNLOAD utility statements.

Values:

    S   FLOAT S390 is added.

    I   FLOAT IEEE is added.

    blank   The FLOAT option is not added; DB2 default utility options are used.

Default:
    blank

util_unload_format_internal

The **util_unload_format_internal** parameter specifies the FORMAT INTERNAL option for generated UNLOAD utility statements.

Values:

    YES   FORMAT INTERNAL is added. Data is unloaded with format internal when applicable.

Default:

util_unload_implicit_tz

The **util_unload_implicit_tz** parameter specifies the IMPLICIT_TZ option for generated UNLOAD utility statements.

Values:

    A valid IMPLICIT_TZ value for UNLOAD
        The IMPLICIT_TZ option is added with the specified value.

    blank   The IMPLICIT_TZ option is not added; DB2 default utility options are used.

Default:
    blank

util_unload_maxerr

The **util_unload_maxerr** parameter specifies the MAXERR option for generated UNLOAD utility statements.

Values:

A valid MAXERR value for UNLOAD
>   The MAXERR option is added with the specified value.

blank   The MAXERR option is not added; DB2 default utility options are used.

>   **Default:**
>   blank

util_unload_mixed_ccsid
>   The **util_unload_mixed_ccsid** parameter specifies the MIXED CCSID option for generated UNLOAD utility statements.
>
>   **Values:**
>
>   **A valid CCSID value**
>   >   The CCSID option is added with the specified value. E.g. CCSID(**util_unload_sbcs_ccsid**, **util_unload_mixed_ccsid**, **util_unload_dbcs_ccsid**)
>
>   **blank**   The value is omitted from the CCSID option; DB2 default utility options are used.
>
>   **Default:**
>   blank

util_unload_nopad
>   The **util_unload_nopad** parameter specifies the NOPAD option for generated UNLOAD utility statements.
>
>   **Values:**
>
>   **Y**   NOPAD is added.
>
>   **N**   The NOPAD option is not added.
>
>   **Default:**
>   N

util_unload_nosubs
>   The **util_unload_nosubs** parameter specifies the NOSUBS option for generated UNLOAD utility statements.
>
>   **Values:**
>
>   **Y**   NOSUBS is added.
>
>   **N**   The NOSUBS option is not added.
>
>   **Default:**
>   N

util_unload_parallel
>   The **util_unload_parallel** specifies the maximum number of subtasks that are to be used in parallel when unloading a partitioned table space.
>
>   **Values:**
>
>   **YES**   The PARALLEL option is added.
>
>   **integer**
>   >   0-32767. The PARALLEL option is added to the utility statement with the specified value.
>
>   **blank**   The PARALLEL option is not added; DB2 default utility options are used.

Default:
>blank

util_unload_sbcs_ccsid
>The **util_unload_sbcs_ccsid** parameter specifies the SBCS CCSID option for generated UNLOAD utility statements.
>
>**Values:**
>>**A valid CCSID value**
>>>The CCSID option is added with the specified value. E.g. CCSID(**util_unload_sbcs_ccsid**, **util_unload_mixed_ccsid**, **util_unload_dbcs_ccsid**)
>>
>>**blank** The value is omitted from the CCSID option; DB2 default utility options are used.
>
>**Default:**
>>blank

util_unload_shrlevel
>The **util_unload_shrlevel** parameter specifies the SHRLEVEL option for generated UNLOAD utility statements.
>
>**Values:**
>>**1** SHRLEVEL CHANGE ISOLATION CS is added.
>>
>>**2** SHRLEVEL CHANGE ISOLATION UR is added.
>>
>>**3** SHRLEVEL REFERENCE is added.
>>
>>**blank** The SHRLEVEL option is not added; DB2 default utility options are used.
>
>**Default:**
>>blank

util_unload_skip_locked_data
>The **util_unload_skip_locked_data** parameter specifies the SKIP LOCKED DATA option for generated UNLOAD utility statements.
>
>**Values:**
>>**YES** SKIP LOCKED DATA is added.
>>
>>**NO** The SKIP LOCKED DATA option is not added.
>
>**Default:**
>>NO

validate_wsl
>The **validate_wsl** parameter specifies whether to validate the WSL after it is created. If the change has prerequisites, this option is forced to NO.
>
>**Values:**
>>**Y** Validate the WSL and display the report in the job output.
>>
>>**N** Do not validate the WSL.
>
>**Default:**
>>N

validation_stmtexit
>The **validation_stmtexit** parameter specifies a REXX exec that will be called to validate a DB2 statement.

Using parameter profiles: Change Management batch interface

The product default parameter values can be overridden. When the Change
Management batch interface is invoked, it reads parameters from the following
two files in sequence: PROFPARM DD, then PARMS DD.

About this task

The Change Management batch interface reads two files for parameters in order to
enable installations to more easily establish, maintain, and use their own default
parameter values. This can be done by putting installation defaults into the
PROFPARM DD and individual invocation overrides into the PARMS DD.

One method for setting up profiles is to define the PROFPARM DD in the JCL
procedure and define the PARMS DD when invoking the JCL procedure. This enables
the JCL procedure parameter (for example, the SSID or the user-customized JCL
procedure parameter) to dynamically determine which data set(s) to associate with
the parameter file in the JCL procedure (PROFPARM DD) .

To use this method, use the following procedure, and refer to the examples that
follow.

Procedure

1. Define the PROFPARM DD in the JCL procedure.
2. Define the PARMS DD when invoking the JCL procedure.

Example 1: Defining the PROFPARM file in the JCL procedure and using the DB2 SSID to determine which parameter profile is used

```
//GOCCM   PROC SSID=,PLAN=,SPCUNIT=SYSDA
//PROFPARM  DD DISP=SHR,DSN=USERID.SSID.PARMS(&SSID)
...
//GOCCM   PEND
```

When the Change Management batch interface is invoked, the SSID parameter
value determines the member name in USERID.SSID.PARMS to use.

Invoking the JCL procedure:
```
//DEMO     JOB (&SYSUID,ICE,ICE,ICE),'DEMO',CLASS=B,
//   MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID,TIME=(,30),
//   REGION=0M
//*
/*JOBPARM S=SY4A
//*
//LSCLIBS JCLLIB ORDER=JCL.PROCLIB
//*
//GOCCM    EXEC GOCCM,SSID=DSNA,PLAN=ADB
//GOCCM.PARMS  DD *
/*
//GOCCM.IMCHG001 DD DISP=SHR,DSN=<DDL OR DELTA CHANGE FILE>
```

The SSID JCL parameter value is DSNA, so the data set name for the `PROFPARM DD` in the JCL procedure resolves to the following:

```
//PROFPARM  DD DISP=SHR,DSN=USERID.SSID.PARMS(DSNA)
```

Any parameter specified in the PARMS DD overrides what is specified in the `PROFPARM DD`.

Example 2: Defining the PROFPARM file in the JCL procedure and using the DB2 SSID to determine which parameter profile is used (same as Example 1). Also, defining a user-customized JCL procedure parameter that determines which additional profile is used

```
//GOCCM    PROC SSID=,PLAN=,SPCUNIT=SYSDA,PROF=EMPTY
//PROFPARM  DD DISP=SHR,DSN=USERID.SSID.PARMS(&SSID)
//          DD DISP=SHR,DSN=USERID.PROF.PARMS(&PROF)
...
//GOCCM    PEND
```

When the Change Management batch interface is invoked, the SSID parameter value determines the member name in USERID.SSID.PARMS to use. The PROF parameter value determines the member name in USERID.PROF.PARMS to use.

Invoking the JCL procedure:

```
//DEMO     JOB (&SYSUID,ICE,ICE,ICE),'DEMO',CLASS=B,
//    MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID,TIME=(,30),
//    REGION=0M
//*
/*JOBPARM S=SY4A
//*
//LSCLIBS JCLLIB ORDER=JCL.PROCLIB
//*
//GOCCM     EXEC GOCCM,SSID=DSNA,PLAN=ADB,PROF=LARGE
//GOCCM.PARMS  DD *
/*
//GOCCM.IMCHG001 DD DISP=SHR,DSN=<DDL OR DELTA CHANGE FILE>
```

The SSID JCL parameter value is DSNA, and the user-defined JCL parameter PROF is LARGE, so the data set names for the `PROFPARM DD` in the JCL procedure resolves to the following:

```
//PROFPARM  DD DISP=SHR,DSN=USERID.SSID.PARMS(DSNA)
//          DD DISP=SHR,DSN=USERID.PROF.PARMS(LARGE)
```

Any parameter specified in the PARMS DD overrides what is specified in the `PROFPARM DD`.

The parameters are read in the following order.

Note: The value for a parameter is the last one read in.
1. USERID.SSID.PARMS(DSNA)
2. USERID.PROF.PARMS(LARGE)
3. The PARMS file

Using symbol variables: Change Management batch interface

Symbol variables provide a method to define patterns for Change Management batch interface parameters related to data set names, new change owner, new change name, and so on.

The date-related and time-related symbol values are refreshed before saving or generating a base version. This enables a time-related variable, such as current timestamp (&CURTS.), to have different values when saving or generating multiple base versions in the same invocation of Change Management batch interface.

Topics:
- "Product-defined symbol variables: Change Management batch interface"
- "Using user-defined symbol variables: Change Management batch interface" on page 712
- Symbol variables in the ADBTEMPL file: DB2 TEMPLATE support

Product-defined symbol variables: Change Management batch interface

The following table lists the product-defined symbol variables available in the Change Management batch interface. The value for each symbol variable is resolved at runtime.

Symbol variables can be specified in all of the Change Management batch interface parameters:

Note: Time-related variables are resolved one time and remain the same value wherever they are used.

Table 20. Product-defined symbol variables for Change Management batch interface

Symbol variable	Description
&SSID. or &SS.	Subsystem ID
&CURSQLID.	CURRENT SQLID
&CURTS.	CURRENT TIMESTAMP
&DATE. or &DT.	YYYYDDD
&JDAY. or &JD.	DDD portion of &DATE.
&JOBNAME. or &JO.	The z/OS job name
&USERID. or &US.	The user ID of the person who is running the job.
&YEAR. or &YE.	YYYY
&MONTH. or &MO.	MM
&DAY. or &DA.	DD
&TIME. or &TI.	HHMMSS
&HOUR. or &HO.	HH portion of &time.
&MINUTE. or &MI.	MM portion of &time.
&SECOND. or &SC.	SS portion of &time.

Table 20. Product-defined symbol variables for Change Management batch interface (continued)

Symbol variable	Description
&CHGTAG.	An identifier that distinguishes between different registered changes on a DB2 subsystem. The **chgtag_type** CM Batch parameter specifies the type of value that &CHGTAG. resolves to: • values that are based on the DB2 Admin generated change ID number. • a user specified change name. • a user specified change owner. When the chgtag_type is ID, the &CHGTAG. symbol variable resolves to values based on the DB2 Admin generated change ID: • For an original change, C(changeid) in data set names. The WSL PDS member is C(changeid) and the run JCL PDS member is E(changeid). • For a recover change, R(changeid) both in data set names, and for the recover job JCL and WSL PDS members. The (changeid) is the numeric change ID of the original change. When the chgtag_type is NAME, the &CHGTAG. symbol variable resolves to values based on the user specified change name: • When processing the original change, the change name for data set names. The member name is the original change name for both the WSL PDS member and the run JCL PDS member. • When processing the recover change, the change name of the original change with '.R' appended for data set names. The member name is the original change name for both the recover WSL PDS member and the recover JCL PDS member. When the chgtag_type is OWNER, the &CHGTAG. symbol variable resolves to values based on the user specified change owner: • When processing the original change, the change owner for data set names. The member name is the original change owner for both the WSL PDS member and the run JCL PDS member. • When processing the recover change, the change owner of the original change with '.R' appended for data set names. The member name is the original change owner for both the recover WSL PDS member and the recover JCL PDS member.

Change tag (&CHGTAG.) Usage: chgtag_type = 'NAME'

Using the change name instead of the change ID can be helpful when you want to automate portions of your change management process, or when you want to more easily locate data sets associated with a change. However, be aware of the restrictions. For example, the change name of the original change must be less than or equal to 8 characters, and conform to z/OS data set naming rules.

Attention: It is critical that you ensure that each registered change on a DB2 subsystem is unique by its change name only. The change name is used in data set names and common PDS member names. There might be two changes with the same change name but with different change owners on the same DB2 subsystem. Failing to ensure unique change names for all registered changes can result in change artifacts in a data set being overwritten by another change with the same change name. DB2 Admin enforces the uniqueness by change owner and change name, but not by the change name alone. If all users of DB2 Admin Change Management use the same change owner, then DB2 Admin ensures that the change name is unique for all registered changes on a DB2 subsystem.

Attention: If the same PDS is used to store change artifacts for multiple DB2 subsystems (for example, the run JCL PDS), you must ensure that the change name is unique across all DB2 subsystems that share the PDS. Take extra care to ensure that you have unique change names across multiple DB2 subsystems.

When a recover change is requested, the PDS member name is the same for both the original and recover change. The following data sets for a recover change must be different from the original change data sets after symbols are resolved:

- pds_for_recover_jcl must be different from pds_for_run_jcl
- pds_for_recover_wsl must be different from pds_for_wsl

There is a change in behavior to the DB2 Admin skeleton template data set name customization, specifically to skeletons ADB2UCUS and ADB2UCUU. When you are generating the recover change, the &LEVEL symbol in the ADB2UCUS and ADB2UCUU skeletons is 2 characters more than the value of the change name of the original change. Because the maximum length of the change name is 8, &LEVEL must be a maximum length of 10, instead of the normal maximum length of 8. For example, if the change_name is ABCDEFGH (character length of 8), then when you are generating the recover change, the &LEVEL resolves to ABCDEFGH.R (a character length of 10).

Change tag (&CHGTAG.) Usage: chgtag_type = 'OWNER'

Use of the change owner instead of the change name provides more flexibility for user customized environments in which the change owner and change name have different meanings. Restrictions and considerations when you specify the chgtag_type as OWNER is similar to chgtag_type as NAME. The change owner of the original change must be less than or equal to 8 characters, and conform to z/OS data set naming rules.

Attention: It is critical that you ensure that each registered change on a DB2 subsystem is unique by its change owner only. The change owner is used in data set names and common PDS member names. There might be two changes with the same change owner but with different change names on the same DB2 subsystem. Failing to ensure unique change owners for all registered changes can result in change artifacts in a data set being overwritten by another change with the same change owner. DB2 Admin enforces the uniqueness by change owner and change name, but not by the change owner alone. If all users of DB2 Admin Change Management use the same change name, then DB2 Admin ensures that the change owner is unique for all registered changes on a DB2 subsystem.

Attention: If the same PDS is used to store change artifacts for multiple DB2 subsystems (for example, the run JCL PDS), you must ensure that the change owner is unique across all DB2 subsystems that share the PDS. Take extra care to ensure that you have unique change owners across multiple DB2 subsystems.

When a recover change is requested, the PDS member name is the same for both the original and recover change. The following data sets for a recover change must be different from the original change data sets after symbols are resolved:

- pds_for_recover_jcl must be different from pds_for_run_jcl
- pds_for_recover_wsl must be different from pds_for_wsl

There is a change in behavior to the DB2 Admin skeleton template data set name customization, specifically to skeletons ADB2UCUS and ADB2UCUU. When you are generating the recover change, the &LEVEL symbol in the ADB2UCUS and ADB2UCUU skeletons is 2 characters more than the value of the change owner of the original change. Because the maximum length of the change name is 8, &LEVEL must be a maximum length of 10, instead of the normal maximum length of 8. For example, if the change_owner is ABCDEFGH (character length of 8), then when you are generating the recover change, the &LEVEL resolves to ABCDEFGH.R (a character length of 10).

&CHGTAG. examples

chgtag_type = 'ID'

When an original change consists of a DB2 Admin generated change ID of 45, and the user specified change name is ABCDEFGH, &CHGTAG. resolves to C0000045, when files are generated for the original change. The run JCL PDS member name is E0000045. When files are generated for the recover change, &CHGTAG. resolves to R0000045. The recover JCL PDS member name is R0000045. Assuming default values are used for the data set names, the following is a subset of the data set names that are used for the original change:

```
USERID.SSID.C0000045.CHG
USERID.SSID.C0000045.IFF
USERID.SSID.RUN.WSL(C0000045)
USERID.SSID.RUN.JCL(E0000045)
USERID.SSID.C0000045.IN
```

The following is a subset of the data set names that are used for the recover change:

```
USERID.SSID.R0000045.CHG
USERID.SSID.R0000045.IFF
USERID.SSID.RECOVER.WSL(R0000045)
USERID.SSID.RECOVER.JCL(R0000045)
USERID.SSID.R0000045.IN
```

| **chgtag_type = 'NAME'**

| When an original change consists of a DB2 Admin generated change ID of 45, and
| the user specified change name is ABCDEFGH, the original change name of
| ABCDEFGH is used as the PDS member name for the JCL and WSL PDS members.
| When generating data set names for the original change, &CHGTAG. resolves to
| the original change name. When generating data set names for the recover change,
| &CHGTAG. resolves to the original change name with '.R' appended. Assuming
| default values are used for the data set names, the following is a subset of the data
| set names that are used for the original change:

```
USERID.SSID.ABCDEFGH.CHG
USERID.SSID.ABCDEFGH.IFF
USERID.SSID.RUN.WSL(ABCDEFGH)
USERID.SSID.RUN.JCL(ABCDEFGH)
USERID.SSID.ABCDEFGH.IN
```

| The following is a subset of the data set names that are used for the recover
| change:

```
USERID.SSID.ABCDEFGH.R.CHG
USERID.SSID.ABCDEFGH.R.IFF
USERID.SSID.RECOVER.WSL(ABCDEFGH)
USERID.SSID.RECOVER.JCL(ABCDEFGH)
USERID.SSID.ABCDEFGH.R.IN
```

| ## Using user-defined symbol variables: Change Management batch interface

You can define your own user-defined symbol variables and values to define
patterns for Change Management batch interface parameters.

About this task

You can use user-defined symbol variables in any parameter that a product-defined
symbol variable can be specified. For a list of parameters that support
product-defined symbol variables, see "Product-defined symbol variables: Change
Management batch interface" on page 708.

Procedure

1. To learn how to use user-defined symbol variables, refer to the following
   examples.

Example 1: Defining the symbol &TASKNUM.

Suppose you define a symbol &TASKNUM. with a value of A123. &TASKNUM.
could be referenced in the parameters like the following:

- prefix_for_data_sets: &USERID..&TASKNUM.
- pds_for_wsl: &SSID..ANALYZE.WSL
- pds_for_jcl: &SSID..ANALYZE.JCL
- new_change_owner: &CURSQLID.
- new_change_name: &TASKNUM.-&CURTS.

```
//DEMO     JOB (&SYSUID,ICE,ICE,ICE),'DEMO',CLASS=B,
//   MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID,TIME=(,30),
//   REGION=0M
//*
/*JOBPARM S=SY4A
//*
//LSCLIBS JCLLIB ORDER=JCL.PROCLIB
//*
```

```
//GOCCM    EXEC GOCCM,SSID=DSNA,PLAN=ADB
//GOCCM.PARMS  DD *
New_change_name = '&TASKNUM.-&CURTS.';
Symbol_name='&TASKNUM.', symbol_value='A123';
/*
//GOCCM.IMCHG001 DD DISP=SHR,DSN=<DDL OR DELTA CHANGE FILE>
```

Example 2: Using PROFPARM, PARMS, and user-defined symbols

In file USERID.SSID.PARMS(DSNA), the following parameter is specified using a user-defined symbol &TASKNUM.:

```
New_change_name = '&TASKNUM.-&CURTS.';
```

In the JCL procedure for Change Management batch interface (GOCCM), the PROFPARM file is defined like the following:

```
//GOCCM    PROC SSID=,PLAN=,SPCUNIT=SYSDA
//PROFPARM  DD DISP=SHR,DSN=USERID.SSID.PARMS(&SSID)
...
//GOCCM    PEND
```

In the call to the Change Management batch interface, the PARMS file is defined and the &TASKNUM. symbol is defined as the work order # A123.

```
//DEMO     JOB (&SYSUID,ICE,ICE,ICE),'DEMO',CLASS=B,
//   MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID,TIME=(,30),
//   REGION=0M
//*
/*JOBPARM S=SY4A
//*
//LSCLIBS JCLLIB ORDER=JCL.PROCLIB
//*
//GOCCM    EXEC GOCCM,SSID=DSNA,PLAN=ADB
//GOCCM.PARMS  DD *
Symbol_name='&TASKNUM.', symbol_value='A123';
/*
//GOCCM.IMCHG001 DD DISP=SHR,DSN=<DDL OR DELTA CHANGE FILE>
```

When the Change Management batch interface is invoked the PROFPARM file gets resolved to:

```
//PROFPARM  DD DISP=SHR,DSN=USERID.SSID.PARMS(DSNA)
```

When a new change is created, the change name is something like A123-2011-11-15-22.52.05.42333.

Symbol variables in the ADBTEMPL file: DB2 TEMPLATE support

You can specify DB2 TEMPLATE statements in the ADBTEMPL file. References to specific symbol variables in the ADBTEMPL file are resolved by DB2 Admin before the template statement is sent to DB2.

References to the following symbol variables are resolved by DB2 Admin:

- User-defined symbol variables that are defined in the CM Batch parameter list.
- The &CHGTAG. product-defined symbol variable.
- The following product-defined symbol variables that are only resolved when referenced in the ADBTEMPL file:

Table 21. Symbol variables that are resolved only when referenced in the ADBTEMPL file

Symbol variable	Description
&PREFIX.	The value of the prefix_for_data_sets parameter.
&TNAME.	Object type count ID. Resolves to a one character object type identifier followed by a count of that object type. The maximum length this symbol variable resolves to is 5. The following examples show the processed symbol variables: T0001 - first table T0002 - second table T0003 - third table and so on. S0001 - first table space S0002 - second table space S0003 - third table space and so on. I0001 - first index I0002 - second index I0003 - third index and so on.

Substring notation is not supported for DB2 Admin and user-defined symbol variables.

Importing changes using the Change Management batch interface

You can use the Change Management batch interface to import changes.

Scenario 1: Importing multi-target changes

To import a DDL file to multiple target locations, you can just run one Change Management batch job that imports the change to all the subsystems at once. The target locations are the DB2 locations of the remote server that you specify on the Insert a Target panel. In this example, the target locations are DSNA and DSNB.

If your subsystems have DRDA connectivity, the changes will be automatically imported across subsystems. If your subsystems do not have DRDA connectivity and are using the FILE method, you must manually import the changes, one subsystem at a time.

Before you begin

You should have a DDL file containing all the object definitions, or a changes file containing the delta change statements that you want to apply to your targets. A sample DDL file is shown below.

```
BROWSE SYSADM.DDL(XDB) - 01.00            Line 0000000000 Col 001 080
Command ===>                                        Scroll ===> CSR
******************************** Top of Data *********************************
  CREATE DATABASE XDB
```

```
     BUFFERPOOL BP0
     INDEXBP    BP1
     CCSID      EBCDIC
     STOGROUP   SYSDEFLT;
******************************* Bottom of Data *******************************
```

You should also have masks defined for your targets, as described in "Creating a mask" on page 742. In this example, the masks *"SYSADM"."MASK A1"* and *"SYSADM"."MASK A2"* are defined on the subsystem DSNA. The mask *"SYSADM"."MASK B"* is defined on the subsystem DSNB.

For example, "SYSADM"."MASK A1" is defined as:

DBNAME :XDB,A1DB

Procedure

1. Ensure that you are on the central system, from which you want to initiate the multi-target change. The central system is used to track the changes that you import to the target systems. In this example, subsystem DSNA is used as the central system.
2. On the Manage Targets panel (ADBPC9), select Option 3 - Insert a target.
3. On the Insert a Target panel (ADBPC911), define your target. You must define each target that you want to import changes to. In this example, "TARGET A1", "TARGET A2", and "TARGET B" are defined, as shown in the figures below.

```
 ADBPC911  --------------- CM - Insert a Target ------------------------- 15:50
 Command ===>


 Type new values and press Enter.

 *Name  . . . . . . . . . TARGET A1         > (? to lookup)
 *DB2 location . . . . . DSNA               > (? to lookup)
 Comment  . . . . . . . .                   >
 *Communication method . DRDA               (DRDA or File)
 Mask owner at target . . SYSADM            >
 Mask name at target  . . MASK A1           >
```

Figure 483. Defining Target A1

```
 ADBPC911  --------------- CM - Insert a Target ------------------------- 15:50
 Command ===>


 Type new values and press Enter.

 *Name  . . . . . . . . . TARGET A2         > (? to lookup)
 *DB2 location . . . . . DSNA               > (? to lookup)
 Comment  . . . . . . . .                   >
 *Communication method . DRDA               (DRDA or File)
 Mask owner at target . . SYSADM            >
 Mask name at target  . . MASK A2           >
```

Figure 484. Defining Target A2

```
ADBPC911  --------------- CM - Insert a Target ------------------------- 15:50
Command ===>


Type new values and press Enter.

*Name . . . . . . . . . . TARGET B              > (? to lookup)
*DB2 location . . . . . DSNB                    > (? to lookup)
Comment . . . . . . . .                         >
*Communication method . DRDA                    (DRDA or File)
Mask owner at target . . SYSADM                 >
Mask name at target . . MASK B                  >
```

Figure 485. Defining Target B

4. Create a Change Management batch JCL. In the Change Management batch file, specify values for the following parameters:

- ACTION_IMPORT_CHANGE = 'Y'. You must set this parameter to Y in order to import.
- CHANGE_OWNER
- CHANGE_NAME
- TARGET_PROFILE_NAME
- TARGET_MASK_OWNER
- TARGET_MASK_NAME
- TARGET_CHANGE_OWNER
- TARGET_CHANGE_NAME

A sample batch file is shown below.

```
<JOB CARDS>
//*
//ADBLIBS JCLLIB ORDER=<CM batch PROCLIB>
//*
//****************************************************************GOCCM*******
//* STEP TO RUN A MULTI-TARGET CHANGE
//****************************************************************************
//ANLYZ EXEC GOCCM,SSID=DSNA,PLAN=ADB
//GOCCM.PARMS  DD *
ACTION_IMPORT_CHANGE='Y'
ACTION_ANALYZE_CHANGE='N'
IMPORT_PENDING_CHANGE_ACTION='S'
CHANGE_OWNER='SYSADM'
CHANGE_NAME='CENTRAL MTC CHANGE - FOR TARGETS A1, A2 & B'
;
TARGET_PROFILE_NAME='TARGET A1'
,TARGET_MASK_NAME='MASK A1'
,TARGET_CHANGE_OWNER='SYSADM'
,TARGET_CHANGE_NAME='CHANGE1 FOR TARGET A1';
TARGET_PROFILE_NAME='TARGET A2'
,TARGET_MASK_OWNER='SYSADM'
,TARGET_MASK_NAME='MASK A2'
,TARGET_CHANGE_OWNER='SYSADM'
,TARGET_MASK_NAME='MASK A2'
,TARGET_CHANGE_OWNER='SYSADM'
,TARGET_CHANGE_NAME='CHANGE2 FOR TARGET A2';
TARGET_PROFILE_NAME='TARGET B'
TARGET_MASK_OWNER='SYSADM'
,TARGET_CHANGE_NAME='CHANGE2 FOR TARGET A2';
TARGET_PROFILE_NAME='TARGET B'
,TARGET_MASK_OWNER='SYSADM'
,TARGET_MASK_NAME='MASK B'
,TARGET_CHANGE_OWNER='SYSADM'
,TARGET_CHANGE_NAME='CHANGE3 FOR TARGET B';
```

```
/*
//IMCHG001 DD DISP=SHR,
//          DSN=SYSADM.DDL(XDB)
//MTC      DD DISP=SHR,
//          DSN=SYSADM.MTC
//          ENDIF
```

5. Run the batch job.

   After the batch job is run successfully, a multi-target import report is generated, as shown below.

```
******************************** TOP OF DATA ********************************
--------------------------------------------------------------------------------
ADB2CID - Import changes - 2015-11-11 15:07
--------------------------------------------------------------------------------


ADB2CID - Multi-Target Change Summary

Multi-target change id:          7913

Target    Owner    Name                Status
--------- -------- ------------------ ----------------------------------------
TARGET A1 SYSADM   CHANGE1 FOR TARGET ADB9400I:The change was registered successfully, Change
TARGET A2 SYSADM   CHANGE2 FOR TARGET ADB9400I:The change was registered successfully, Change
TARGET B  SYSADM   CHANGE3 FOR TARGET ADB9400I:The change was registered successfully, Change

ADB2CID - Multi-Target Change End of Summary

ADB0004I ADBCCM - Ended normally
******************************** BOTTOM OF DATA ********************************
```

What to do next

After importing your changes, you may want to verify that your imports were successful. You can issue the AT line command to check the status of your imports.

1. On the Change Management panel (ADB2C), select option 1 - Manage changes.
2. On the Manage Changes panel (ADB2C1), select option 1 - Display changes.
3. On the Changes panel (ADB2C11), issue the AT line command for the multi-target change to see the status of the target changes, as displayed in the figure below.

```
ADB2C11 n ------------------ CM - Changes -------------------- Row 1 to 3 of 3
Command ===>                                            Scroll ===> CSR

Commands: COMMENT  EXPORT
Line commands:
 U - Update  AN - Analyze  RN - Run  VE - Versions  ST - Statements
 PQ - Prerequisites  IG - Ignores  MA - Masks  S - Show WSL  B - Checkpoint
 AT - Associated Targets  ? - Show all line commands

Sel      ID Owner    Name                      Type     Status   I Comment
         * *          *                        *        *        * *
--- -------- -------- -------------------------> -------- -------- - ---------
       7915 SYSADM   CHANGE2 FOR TARGET A2      CHANGE   DEFINED
       7914 SYSADM   CHANGE1 FOR TARGET A1      CHANGE   DEFINED
AT     7913 SYSADM   CENTRAL MTC CHANGE - FOR TA MULTI-TC DEFINED
****************************** END OF DB2 DATA ******************************
```

Figure 486. Changes panel (ADB2C11)

The Associate Targets panel (ADBPCMT) is displayed, as shown below. The Associate Targets panel shows the targets that your change was successfully

imported to.

```
ADBPCMT n --------------- CM - Associate Targets ------------- Row 1 to 3 of 3
Command ===>                                              Scroll ===> CSR

Details for multi-target change: SYSADM.CENTRAL MTC   > DB2 System: DSNA
                                                        DB2 SQL ID: SYSADM
Line commands:
 I - Interpret

    Target                Change   Change
Sel Name    DB2 Location  Owner    Name                       Status
    *       *             *        *                          *
--- ------->  ------------> --------- ---------------------- -------------------
    TARGET A DSNA         SYSADM   CHANGE1 FOR TARGET A1  DEFINED
    TARGET A DSNA         SYSADM   CHANGE2 FOR TARGET A2  DEFINED
    TARGET B DSNB         SYASDM   CHANGE3 FOR TARGET B   DEFINED
***************************** END OF DB2 DATA *******************************
```

Figure 487. Associate Targets panel (ADBPCMT)

Scenario 2: Importing changes when your subsystems are not connected

To import the same DDL or change file to unconnected subsystems DSNA, DSNB, and DSNC, you will need to run a Change Management batch JCL for each subsystem.

In general, if the JCL procedure has been set up so that the SSID parameter determines the DB2 libraries for the subsystem, the same CM batch JCL procedure can be used to import the change into the different subsystems.

Procedure

The following examples show how to import a change into multiple DB2 subsystems, one subsystem at a time.

Example 1: Import to DSNA
```
//DEMO     JOB (&SYSUID,ICE,ICE,ICE),'DEMO',CLASS=B,
//   MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID,TIME=(,30),
//   REGION=0M
//*
/*JOBPARM S=SY4A
//*
//LSCLIBS JCLLIB ORDER=JCL.PROCLIB
//*
//GOCCM    EXEC GOCCM,SSID=DSNA,PLAN=ADB
//GOCCM.PARMS  DD *
  Symbol_name='&TASKNUM.', symbol_value='A123';
/*
//GOCCM.IMCHG001 DD DISP=SHR,DSN=DELTA.CHANGE(A123)
```

Example 2: Import to DSNB
```
//DEMO     JOB (&SYSUID,ICE,ICE,ICE),'DEMO',CLASS=B,
//   MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID,TIME=(,30),
//   REGION=0M
//*
/*JOBPARM S=SY4A
//*
//LSCLIBS JCLLIB ORDER=JCL.PROCLIB
//*
//GOCCM    EXEC GOCCM,SSID=DSNB,PLAN=ADB
//GOCCM.PARMS  DD *
```

```
   Symbol_name='&TASKNUM.', symbol_value='A123';
/*
//GOCCM.IMCHG001 DD DISP=SHR,DSN=DELTA.CHANGE(A123)
```

Example 3: Import to DSNC

```
//DEMO      JOB (&SYSUID,ICE,ICE,ICE),'DEMO',CLASS=B,
//   MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID,TIME=(,30),
//   REGION=0M
//*
/*JOBPARM S=SY4A
//*
//LSCLIBS JCLLIB ORDER=JCL.PROCLIB
//*
//GOCCM     EXEC GOCCM,SSID=DSNC,PLAN=ADB
//GOCCM.PARMS  DD *
  Symbol_name='&TASKNUM.', symbol_value='A123';
/*
//GOCCM.IMCHG001 DD DISP=SHR,DSN=DELTA.CHANGE(A123)
```

Analyzing a multi-target change

You can analyze changes applied to multiple targets using Change Management
batch.

About this task

You can analyze all the target changes that are registered locally on the central
system by analyzing the multi-target change using the Change Management batch
interface. Analyzing changes in batch is a more efficient alternative to analyzing
changes one at a time through the Change Management panels.

Procedure

1. In the Change Management batch file, specify ACTION_ANALYZE_CHANGE
   = 'Y', as shown below. You can specify additional CM batch parameters as
   needed.
   ```
   <JOB CARDS>
   //*
   //ADBLIBS JCLLIB ORDER=<CM batch PROCLIB>
   //*
   //*********************************************************GOCCM*******
   //* STEP TO RUN A MULTI-TARGET CHANGE
   //**********************************************************************
   //ANLYZ EXEC GOCCM,SSID=DSNA,PLAN=ADB
   //GOCCM.PARMS  DD *
    ACTION_IMPORT_CHANGE='N'
    ACTION_ANALYZE_CHANGE='Y'
    CHANGE_OWNER='SYSADM'
    CHANGE_NAME='CENTRAL MTC CHANGE - FOR TARGETS A & B'
   /*
   ```
   Currently, the following parameters are not supported when analyzing a
   multi-target change:
   - RUN_SQLID
   - OBJECT_GRANTOR
   - GENERATE_RECOVER_CHANGE
   - DO_RUNTIME_ANALYZE.

   If you have questions about these restrictions, please contact IBM Support.
2. Run the batch job. After the batch job is run successfully, a multi-target analyze
   report is generated, as shown below.

```
--------------------------------------------------------------------------------
ADBCCM - Multi-target Change Summary Report (Analyze Process)
--------------------------------------------------------------------------------


Multi-target change id: 7913

Successful entries:

Location: DSNA

Target     ID     Owner    Name                         Status   Remarks
---------- ------ -------- ---------------------------- -------- ---------------
TARGET A1  7914   SYSADM   CHANGE1 FOR TARGET A1        ANALYZED Analyze Successful
TARGET A2  7915   SYSADM   CHANGE2 FOR TARGET A2        ANALYZED Analyze Successful


Skipped or Failed or Running entries:

Target     Loc    ID     Owner    Name                  Status   Remarks
---------- ------ ------ -------- --------------------- -------- --------------
TARGET B   DSNB   7915   SYSADM   CHANGE3 FOR TARGET B  DEFINED  Skipped.
                                                                 Target not on local subsystem.


--------------------------------------------------------------------------------
ADBCCM - End of Multi-target Change Summary Report (Analyze Process)
--------------------------------------------------------------------------------


ADB0004I ADBCCM - Ended normally
****************************** BOTTOM OF DATA ********************************
```

In this example, TARGET B was skipped because it is not on the local
subsystem, DSNA. You must use subsystem DSNB to analyze the change for
TARGET B.

What to do next

After analyzing your changes, you may want to verify that your analyze was
successful.

1. On the Change Management panel (ADB2C), select option 1 - Manage changes.
2. On the Manage Changes panel (ADB2C1), select option 1 - Display changes.
3. On the Changes panel (ADB2C11), issue the AT line command for the
   multi-target change.

   The Associate Targets panel (ADBPCMT) is displayed, as shown below. The
   Associate Targets panel shows the status of all your target changes.

```
ADBPCMT n -------------- CM - Associate Targets ------------- Row 1 to 3 of 3
Command ===>                                              Scroll ===> CSR

Details for multi-target change: SYSADM.CENTRAL MTC   > DB2 System: DSNA
                                                        DB2 SQL ID: SYSADM
Line commands:
 I - Interpret

    Target                  Change    Change
Sel Name     DB2 Location   Owner     Name                  Status
    *        *              *         *                     *
--- -------> ------------> --------- --------------------- -------------------
    TARGET A DSNA           SYSADM    CHANGE1 FOR TARGET A1  ANALYZED
    TARGET A DSNA           SYSADM    CHANGE2 FOR TARGET A2  ANALYZED
    TARGET B DSNB           SYSADM    CHANGE3 FOR TARGET B   DEFINED
***************************** END OF DB2 DATA ******************************
```

Figure 488. Associate Targets panel (ADBPCMT)

Running a multi-target change

You can run changes applied to multiple targets using Change Management batch.

About this task

You can run all the target changes that are registered locally on the central system by running the multi-target change using the Change Management batch interface. Running changes in batch is a more efficient alternative to running changes one at a time through the Change Management panels.

Procedure

1. In the Change Management batch file, specify ACTION_RUN_CHANGE = 'Y', as shown below. You can specify additional CM batch parameters as needed.

```
<JOB CARDS>
//*
//ADBLIBS JCLLIB ORDER=<CM batch PROCLIB>
//*
//***********************************************************GOCCM*******
//* STEP TO RUN A MULTI-TARGET CHANGE
//**********************************************************************
//ANLYZ EXEC GOCCM,SSID=DSNA,PLAN=ADB
//GOCCM.PARMS  DD *
  ACTION_IMPORT_CHANGE='N'
  ACTION_ANALYZE_CHANGE='N'
  ACTION_RUN_CHANGE='Y'
  CHANGE_OWNER='SYSADM'
  CHANGE_NAME='CENTRAL MTC CHANGE - FOR TARGETS A & B'
 /*
```

2. Run the batch job. After the batch job is run successfully, a multi-target run report is generated, as shown below.

```
--------------------------------------------------------------------------------
ADBCCM - Multi-target Change Summary Report (Run Process)
--------------------------------------------------------------------------------

Multi-target change id: 7913

Successful entries:

Location: DSNA

Target      ID    Owner    Name                            Status    Remarks
```

```
          ---------- ------ -------- ---------------------------- -------- ---------------
          TARGET A1  7914   SYSADM   CHANGE1 FOR TARGET A1         COMPLETE Run Successful
          TARGET A2  7915   SYSADM   CHANGE2 FOR TARGET A2         COMPLETE Run Successful


          Skipped or Failed or Running entries:

          Target     Loc    ID     Owner    Name                  Status   Remarks
          ---------- ------ ------ -------- -------------------- -------- ---------------

          TARGET B   DSNB   7915   SYSADM   CHANGE3 FOR TARGET B  ANALYZED Skipped. Target is not on local


          ------------------------------------------------------------------------------
          ADBCCM - End of Multi-target Change Summary Report (Run Process)
          ------------------------------------------------------------------------------
```

What to do next

After running your changes, you may want to verify that your run was successful.

1. On the Change Management panel (ADB2C), select option 1 - Manage changes.
2. On the Manage Changes panel (ADB2C1), select option 1 - Display changes.
3. On the Changes panel (ADB2C11), issue the AT line command for the multi-target change.

   The Associate Targets panel (ADBPCMT) is displayed, as shown below. The Associate Targets panel shows the status of all your target changes.

```
ADBPCMT n --------------- CM - Associate Targets ------------- Row 1 to 3 of 3
Command ===>                                              Scroll ===> CSR

Details for multi-target change: SYSADM.CENTRAL MTC   > DB2 System: DSNA
                                                        DB2 SQL ID: SYSADM
Line commands:
 I - Interpret

     Target                 Change    Change
Sel Name     DB2 Location   Owner     Name                     Status
     *        *             *         *                         *
--- -------> ------------> --------- ---------------------- -------------------
    TARGET A DSNA           SYSADM    CHANGE1 FOR TARGET A1  COMPLETE
    TARGET A DSNA           SYSADM    CHANGE2 FOR TARGET A2  COMPLETE
    TARGET B DSNB           SYSADM    CHANGE3 FOR TARGET B   DEFINED
****************************** END OF DB2 DATA ******************************
```

Figure 489. Associate Targets panel (ADBPCMT)

Using DB2 templates: Change Management batch interface

Managing templates when using the Change Management batch interface is done by specifying the DB2 TEMPLATE statement in ADBTEMPL DD. This enables installations to define a data set with DB2 TEMPLATE statements and to easily use these template statements in multiple DB2 subsystems.

About this task

Some Change Management batch interface parameters control whether user-provided templates or product default templates are used while others enable templates for utility type files.

Procedure

In the `ADBTEMPL` file, define each template on a separate line. Depending on how you want to use templates, use the procedure in one of the following options.

Important: The first two words of a template statement must be TEMPLATE followed by the template name, with no SQL comments in between the first two words.

- To control whether user-provided templates or product default templates are used, use the following settings in the **generate_templates** parameter:
  - **Y**: If the `generate_templates` parameter is set to Y, the use of TEMPLATEs is enabled. If you enable templates for a template type individually (for example: util_template_copyddn1_use = 'S', user templates are used for that template type if it is defined in the ADBTEMPL file. If you do not enable templates for a template type, default templates are used for that template type.
  - **N**: If the `generate_templates` parameter is set to N, this allows you to easily disable the use of user-specified TEMPLATEs without having to toggle off or on each template type individually. When the `generate_templates` parameter is set to N, the results is that product default templates are used when templates are needed.
- To enable and make available templates for each utility file type, use the parameter names starting with **util_template** and **util_clone_template**.

  Note: The parameter names starting with **util_clone_template** define the templates used when processing a table space that has a clone table.
  For full descriptions of parameter names starting with **util_template** and **util_clone_template**, see "Parameter definitions: Change Management batch interface" on page 604.

Example

In the following example, templates COPY1 and LOBC are specified in the `ADBTEMPL DD`. The template named COPY1 is the product default template name for the first COPY data set. The template named LOBC is the product default template name for templates associated with LOB columns. The **generate_templates** parameter is set to Y, so these templates are used.

Note: The `ADBTEMPL` file is not processed to resolve product-specific and user-defined variables. The template statements are passed as is to DB2.

```
//LSCLIBS JCLLIB ORDER=GOCA20.SGOCSAMP
//*
//GOCCM     EXEC GOCCM,SSID=DSNA,PLAN=ADB
//GOCCM.PARMS DD *
 CHANGE_NAME = 'AUTO:2011-12-11-15.24.28.803388'
ACTION_ANALYZE_CHANGE = 'Y'
generate_templates = 'Y'
take_an_image_copy = 'B'
run_reorg_rebuild = 'A'
run_check_data = 'Y'
prefix_for_data_sets = '&USERID..&ABC.'
;
 symbol_name = '&ABC.',symbol_value='TMPL';
/*
//ADBTEMPL DD *
TEMPLATE COPY1
```

```
  DSN 'DEMBIN2.TMPL.&SSID..COPY1.&UQ.'
TEMPLATE LOBC
  DSN 'DEMBIN2.TMPL.&SSID..LOB.&UQ.'
/*
```

- If the **generate_templates** parameter is set to N, the templates in ADBTEMPL DD are not used.
- If the **generate_templates** parameter is set to Y and the **util_template_copyddn1_name** parameter is set to ZZZ, the COPY1 template is not used for the first COPY data set because template ZZZ is not defined in the ADBTEMPL DD. In this case, a product default template is used. The LOBC template is still used whenever a template is needed for LOB columns.
- If the **generate_templates** parameter is set to Y, and **util_template_copyddn1_use** is set to ", the COPY1 template is not used for the first COPY data set because user-specified templates is disabled. The LOBC template is still used whenever a template is needed for LOB columns.

Examples: Invoking the Change Management batch interface for various actions

The following examples provide details about using the Change Management batch interface to performs various actions.

Note: For each of these examples, the PROFPARM file in the GOCCM JCL procedure contains the following parameter values:

```
JOB_PARM_LINE_1='S=SY4A'
JOB_JCLLIB_LINE_1='//LSCLIBS JCLLIB ORDER=ADB.DEVCUST.SAMP';
```

- "Example 1: Importing a mask using the default mask name"
- "Example 2: Importing a mask using a user-provided mask name" on page 725
- "Example 3: Importing an ignore" on page 725
- "Example 4: Importing a DDL file using the default change name. The change is imported but not analyzed." on page 725
- "Example 5: Importing more than one delta change file into a single change, and use a user provided change name. The change is imported but not analyzed." on page 726
- "Example 6: Analyze a change. " on page 726
- "Example 7: Run a change." on page 727
- "Example 8: Recover a change" on page 727
- "Example 9: Import, analyze, and build a run job in one invocation of CM batch" on page 728
- "Example 10: Import, analyze, build a run job, and run the change in one invocation of Change Management batch interface" on page 728
- "Example 11: Run compare and register a change to implement the differences" on page 729
- "Example 12: Run compare (same as example 11 but without registering a change)" on page 730
- "Example 13: Run compare, and do not register a change" on page 730

Example 1: Importing a mask using the default mask name

```
//IMMASK   JOB (&SYSUID),'DEMO',CLASS=A,
//   MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID,
//   REGION=0M
//*
/*JOBPARM S=SY4A
```

```
//*
//LSCLIBS JCLLIB ORDER=ADB.DEVCUST.SAMP
//*
//CMBATCH  EXEC GOCCM,SSID=DSNA,PLAN=ADB
//IMMASK   DD *
SGNAME:*,SYSDEFLT
/*
```

Once this job completes, a CM mask exists and is ready for use. The mask owner and name are something like:

```
MASK_OWNER = 'USER123'
MASK_NAME  = 'AUTO:2012-02-10-09.02.06.840242'
```

Example 2: Importing a mask using a user-provided mask name

```
//IMMASK   JOB (&SYSUID),'DEMO',CLASS=A,
//    MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID,
//    REGION=0M
//*
/*JOBPARM S=SY4A
//*
//LSCLIBS JCLLIB ORDER=ADB.DEVCUST.SAMP
//*
//CMBATCH  EXEC GOCCM,SSID=DSNA,PLAN=ADB
//PARMS  DD *
MASK_NAME = 'PROD_SCHEMA'
/*
//IMMASK DD *
SCHEMA:TEST*,PROD*
/*
```

Once this job completes, a CM mask exists and is ready for use. The mask owner and name are something like:

```
MASK_OWNER = 'USER123'
MASK_NAME  = 'PROD_SCHEMA'
```

Example 3: Importing an ignore

```
//IMIGNORE JOB (&SYSUID),'DEMO',CLASS=A,
//    MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID,
//    REGION=0M
//*
/*JOBPARM S=SY4A
//*
//LSCLIBS JCLLIB ORDER=ADB.DEVCUST.SAMP
//*
//CMBATCH  EXEC GOCCM,SSID=DSNA,PLAN=ADB
//IMIGNORE DD *
BPOOL
STGROUP
/*
```

Once this job completes, a CM ignore exists and is ready for use. The ignore owner and name are something like:

```
IGNORE_OWNER = 'USER123'
IGNORE_NAME  = 'AUTO:2012-02-10-09.02.06.840242'
```

Example 4: Importing a DDL file using the default change name. The change is imported but not analyzed.

```
//IMDDL    JOB (&SYSUID),'DEMO',CLASS=A,
//    MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID,
//    REGION=0M
//*
/*JOBPARM S=SY4A
```

```
//*
//LSCLIBS JCLLIB ORDER=ADB.DEVCUST.SAMP
//*
//CMBATCH  EXEC GOCCM,SSID=DSNA,PLAN=ADB
//PARMS    DD *  ACTION_ANALYZE_CHANGE = 'N'
/*
//IMCHG001 DD *
-- FIRST LINE OF DDL FILE MUST BE A SIMPLE COMMENT!
CREATE TABLE IMPORT_DDL_DEMO1 (C1 INT);
CREATE TABLE IMPORT_DDL_DEMO2 (C1 INT);
/*
```

Once this job completes, a CM change exists and is ready for analyze. The change owner and name are something like:

```
CHANGE_OWNER = 'USER123'
CHANGE_NAME  = 'AUTO:2012-02-10-09.02.06.840242'
```

Example 5: Importing more than one delta change file into a single change, and use a user provided change name. The change is imported but not analyzed.

```
//IMCHG    JOB (&SYSUID),'DEMO',CLASS=A,
//   MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID,
//   REGION=0M
//*
/*JOBPARM S=SY4A
//*
//LSCLIBS JCLLIB ORDER=ADB.DEVCUST.SAMP
//*
//CMBATCH  EXEC GOCCM,SSID=DSNA,PLAN=ADB
//PARMS    DD *
CHANGE_NAME = 'W023:&CURTS.'
CHANGE_COMMENT = 'THIS CHANGE IS FOR WORK ITEM W023.'
ACTION_ANALYZE_CHANGE = 'N'
/*
//IMCHG001 DD DISP=SHR,DSN=USER123.CMDEMOB.W001.DCHG
//IMCHG002 DD DISP=SHR,DSN=USER123.CMDEMOB.W002.DCHG
```

Tip: Instead of hard coding the work order number W023 in multiple places, use a user-defined symbol variable like the following.

```
//PARMS    DD *
 CHANGE_NAME = '&WORK#.:&CURTS.'
CHANGE_COMMENT = 'THIS CHANGE IS FOR WORK ITEM &WORK#..'
ACTION_ANALYZE_CHANGE = 'N'
symbol_name = '&WORK#.',
symbol_value = 'W023';
/*
```

Once this job completes, a CM change exists and is ready for analyze. The change owner and name are something like:

```
CHANGE_OWNER = 'USER123'
CHANGE_NAME  = 'W023:2012-02-10-09.25.43.232422'
```

Example 6: Analyze a change.

```
//ANCHG    JOB (&SYSUID),'DEMO',CLASS=A,
//   MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID,
//   REGION=0M
//*
/*JOBPARM S=SY4A
//*
//LSCLIBS JCLLIB ORDER=ADB.DEVCUST.SAMP
//*
//CMBATCH  EXEC GOCCM,SSID=DSNA,PLAN=ADB
```

```
//PARMS    DD *
CHANGE_OWNER='USER123'
CHANGE_NAME='W023:2012-02-10-09.25.43.232422'
ACTION_ANALYZE_CHANGE = 'Y'
/*
```

Tip: The **change_owner** and **change_name** parameters were manually copied from the job output that imported the change. Here is an example snippet of the job output:

```
==========================
Detailed change information
==========================
For convenience, the change owner and name are displayed below using
the change management batch parameter syntax:
CHANGE_OWNER='USER123'
CHANGE_NAME='W023:2012-02-10-09.25.43.232422'
```

Once this job completes, the change is in 'ANALYZED' state and ready to be run.

Example 7: Run a change.

To run a change, submit the run job that was generated by Change Management batch interface. View the job output that analyzed the change to determine the location of the run job. For example, the run job location is listed for 'Run job DSN':

```
==========================
Detailed change information
==========================
For convenience, the change owner and name are displayed below using
the change management batch parameter syntax:
CHANGE_OWNER='USER123'
CHANGE_NAME='W023:2012-02-10-09.25.43.232422'

Change ID . . . : 3075
Status . . . . . : ANALYZED
Created by . . . : USER123
Created  . . . . : 2012-02-10-09.25.44.796997
Last altered by  : USER123
Last altered . . : 2012-02-10-09.29.20.253278
Change type  . . : CHANGE
WSL DSN  . . . . : 'USER123.DSNA.ANALYZE.WSL(C0003075)'
Run job DSN  . . : 'USER123.DSNA.RUN.JCL(E0003075)'
Recover job DSN  : 'USER123.DSNA.RUN.JCL(R0003075)'
```

Submit the `'USER123.DSNA.RUN.JCL(E0003075)'` job to run the change. Once this job completes, the change is 'COMPLETE' which means the change was applied to DB2.

Example 8: Recover a change

To recover a change, submit the recover job that was generated by Change Management batch interface. View the job output that analyzed or ran the change to determine the location of the recover job. For example, the recover job location is listed for 'Recover job DSN':

```
==========================
Detailed change information
==========================
For convenience, the change owner and name are displayed below using
the change management batch parameter syntax:
CHANGE_OWNER='USER123'
CHANGE_NAME='W023:2012-02-10-09.25.43.232422'
```

```
Change ID  . . . . : 3075
Status . . . . . . : ANALYZED
Created by . . . : USER123
Created  . . . . : 2012-02-10-09.25.44.796997
Last altered by  : USER123
Last altered . . : 2012-02-10-09.29.20.253278
Change type  . . : CHANGE
WSL DSN  . . . . : 'USER123.DSNA.ANALYZE.WSL(C0003075)'
Run job DSN  . . : 'USER123.DSNA.RUN.JCL(E0003075)'
Recover job DSN  : 'USER123.DSNA.RUN.JCL(R0003075)'
```

Submit the 'USER123.DSNA.RUN.JCL(R0003075)' job to recover the change. Once this job completes, the change is recovered. The change status is set back to 'DEFINED'.

Example 9: Import, analyze, and build a run job in one invocation of CM batch

```
//IMCHG    JOB (&SYSUID),'DEMO',CLASS=A,
//  MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID,
//  REGION=0M
//*
/*JOBPARM S=SY4A
//*
//LSCLIBS JCLLIB ORDER=ADB.DEVCUST.SAMP
//*
//CMBATCH  EXEC GOCCM,SSID=DSNA,PLAN=ADB
//PARMS    DD *
/*
//IMCHG001 DD *
-- FIRST LINE OF DDL FILE MUST BE A SIMPLE COMMENT!
 CREATE TABLE IMPORT_DDL_DEMO3 (C1 INT);
/*
//IMCHG002 DD *
-- FIRST LINE OF DDL FILE MUST BE A SIMPLE COMMENT!
ALTER  TABLE IMPORT_DDL_DEMO3
 ADD COLUMN C2 INT;
/*
```

Note: A delta change file could have been specified for the IMCHG001 and IMCHG002 files instead of specifying DDL.

Once this job completes, a CM change exists and is ready to run. The change status is 'ANALYZED'. The change owner and name are something like:

```
CHANGE_OWNER = 'USER123'
CHANGE_NAME  = 'AUTO:2012-02-10-09.26.33.236111'
```

Example 10: Import, analyze, build a run job, and run the change in one invocation of Change Management batch interface

```
//IMCHG    JOB (&SYSUID),'DEMO',CLASS=A,
//  MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID,
//  REGION=0M
//*
/*JOBPARM S=SY4A
//*
//LSCLIBS JCLLIB ORDER=ADB.DEVCUST.SAMP
//*
//CMBATCH  EXEC GOCCM,SSID=DSNA,PLAN=ADB
//PARMS    DD *
ACTION_RUN_CHANGE = 'Y'
/*
//IMCHG001 DD *
-- FIRST LINE OF DDL FILE MUST BE A SIMPLE COMMENT!
CREATE TABLE IMPORT_DDL_DEMO4 (C1 INT);
/*
```

```
//IMCHG002 DD *
-- FIRST LINE OF DDL FILE MUST BE A SIMPLE COMMENT!
ALTER  TABLE IMPORT_DDL_DEMO4
 ADD COLUMN C2 INT;
/*
```

Note: A delta change file could have been specified for the IMCHG001 and IMCHG002 files instead of specifying DDL.

Once this job completes, a CM change exists and is applied to DB2. The change status is 'COMPLETE'. The change owner and name are something like:

```
CHANGE_OWNER = 'USER123'
CHANGE_NAME  = 'AUTO:2012-02-10-09.26.36.636543'
```

Example 11: Run compare and register a change to implement the differences

The compare source is DDL and the compare target is from the DB2 catalog where the DB2 objects are automatically selected based on the content of the source.

```
//IMCHG    JOB (&SYSUID),'DEMO',CLASS=A,
//   MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID,
//   REGION=0M
//*
/*JOBPARM S=SY4A
//*
//LSCLIBS JCLLIB ORDER=ADB.DEVCUST.SAMP
//*
//* INSERT NEW COLUMN NEWCOL INTO TABLE CMBSAMP.TB01
//*
//CMBATCH  EXEC GOCCM,SSID=DSNA,PLAN=ADB
//PARMS    DD *
   ACTION_COMPARE = 'Y'
/*
//SRCIN DD *
SET CURRENT SQLID = 'DEMBIN2';
CREATE DATABASE CMBSAMP;
COMMIT;
CREATE TABLESPACE CMBSAMP IN CMBSAMP
   MAXPARTITIONS 10;
COMMIT;
CREATE TABLE CMBSAMP.TB01
(C1 INT NOT NULL WITH DEFAULT
,NEWCOL  INT NOT NULL WITH DEFAULT
,C3 INT NOT NULL WITH DEFAULT)
IN CMBSAMP.CMBSAMP;
CREATE INDEX CMBSAMP.TB01IX01
  ON CMBSAMP.TB01 (C1);
CREATE VIEW CMBSAMP.VW01 (C1,C3) AS
  SELECT C1,C3 FROM CMBSAMP.TB01;
/*
```

The job output contains the compare report, and message ADB9917I that lists the location of the output version files and of the DB2 Admin delta change file.

```
ADB9917I Compare data set information:
         Delta change data set name:
            DSN=DEMBIN2.SAMP11.OC.D2013127.T132255.DELTA

         Source version:
           Type  . . : FILE
           Owner . . :
           Name  . . : DEMBIN2.SAMP11.OC.D2013127.T132255.SRCVF

         Target version:
```

```
Type  . . : FILE
Owner . . :
Name  . . : DEMBIN2.SAMP11.OC.D2013127.T132255.TGTVF
```

Example 12: Run compare (same as example 11 but without registering a change)

Set **action_import_change** = 'N'.

```
//IMCHG    JOB (&SYSUID),'DEMO',CLASS=A,
//   MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID,
//   REGION=0M
//*
/*JOBPARM S=SY4A
//*
//LSCLIBS JCLLIB ORDER=ADB.DEVCUST.SAMP
//*
//* INSERT NEW COLUMN NEWCOL INTO TABLE CMBSAMP.TB01
//*
//CMBATCH  EXEC GOCCM,SSID=DSNA,PLAN=ADB
//PARMS    DD *
  ACTION_COMPARE = 'Y'
  ACTION_IMPORT_CHANGE = 'N'
/*
//SRCIN DD *
SET CURRENT SQLID = 'DEMBIN2';
CREATE DATABASE CMBSAMP;
COMMIT;
CREATE TABLESPACE CMBSAMP IN CMBSAMP
  MAXPARTITIONS 10;
COMMIT;
CREATE TABLE CMBSAMP.TB01
(C1 INT NOT NULL WITH DEFAULT
,NEWCOL  INT NOT NULL WITH DEFAULT
,C3 INT NOT NULL WITH DEFAULT)
IN CMBSAMP.CMBSAMP;
CREATE INDEX CMBSAMP.TB01IX01
  ON CMBSAMP.TB01 (C1);
CREATE VIEW CMBSAMP.VW01 (C1,C3) AS
  SELECT C1,C3 FROM CMBSAMP.TB01;
/*
```

The job output contains the compare report, and message ADB9917I as described in example 11.

Example 13: Run compare, and do not register a change

The compare source and target is a user-provided list of DB2 object names, and masking is specified.

```
//IMCHG    JOB (&SYSUID),'DEMO',CLASS=A,
//   MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID,
//   REGION=0M
//*
/*JOBPARM S=SY4A
//*
//LSCLIBS JCLLIB ORDER=ADB.DEVCUST.SAMP
//*
//CMBATCH  EXEC GOCCM,SSID=DSNA,PLAN=ADB
//PARMS    DD *
  ACTION_COMPARE = 'Y'
  ACTION_IMPORT_CHANGE = 'N'
  SOURCE_TYPE = 'USER'
  TARGET_TYPE = 'USER'
/*
//SRCIN DD *
```

```
TYPE='DB' NAME='DBTV2';
//TGTIN DD *
TYPE='DB' NAME='DBTV1';
//MASKS DD *
DBNAME:DBTV2,DBTV1
SCHEMA:SCTV2,SCTV1
/*
```

The job output contains the compare report, and message ADB9917I as described in example 11.

Recovering a change made through Change Management

You can recover changes that have been made through Change Management.

About this task

Changes must be backed out one at a time.

The following criteria must be met to recover a change:
- The change must be in COMPLETE status.
- A recover change must exist for the change and be in ANALYZED status. That is, when the change was analyzed, it was specified that a recover change be created. The WSL that was generated for the recover change during the analyze process must also be available.
- All completed changes that must be recovered first have been recovered. For example, assume that you made the following changes:
  1. Created a table space.
  2. Created a table in the table space.
  3. Modified the table to insert a new column.

  If you want to recover the change that created the table space, which would be to drop the table space, you must first recover the change to insert the new column into the table and then recover the change to create the table. Each of these changes must have a recover change.

  When you attempt to recover a change, DB2 Admin automatically identifies any completed changes that must be recovered first and lists them in the order in which you need to recover them. The list of changes represents those changes that have completed after the change to be recovered completed and that modify the same or a related set of objects in the change to be recovered.

To recover a change:

Procedure

1. Display the change to be recovered by selecting option 1 on the Change Management panel and then option 1 on the Manage Changes panel.
2. Issue the RC line command for the change that you want to recover.

   Important: You always recover a change by issuing the recover line command (RC) for the change to recover. You cannot issue the run line command (RN) for the recover change itself.

   DB2 Admin will prompt you in the following situations:
   - If the change cannot be recovered because it has no recover change (or change that must be recovered first does not have a recover change), an error message is issued.

- If the change cannot be recovered because other changes must be recovered first, a panel is displayed with the list of changes that must be recovered first and the order in which the changes must be recovered. Recover the list of changes in the order that is specified before you recover this change.

  The following figure shows an example of the panel that might be displayed when other changes need to be recovered first.

```
DB2 Admin ---------------- DB2X CM - Recover Strategy ----------- Row 1 from 4
Command ===>                                             Scroll ===> PAGE


Recover strategy for change "JOHNSON"."CR_HRDB"
Line commands:
 CH - Change  I - Interpret

    Rcvr
Sel Order Owner    Name               Statement
        * *        *                  *
--- ----- -------- ------------------ ------------------------------------------->
        1 JOHNSON  CR_HRDEPT          CREATE TABLE HRDEPT (DEPTNO CHAR(3) NOT N
        2 JOHNSON  CR_HREMP           CREATE TABLE HREMP (EMPNO CHAR(6)) IN HRD
        3 JOHNSON  CR_HRTS2           CREATE TABLESPACE HRTS2 IN HRDB
        4 JOHNSON  CR_HRTS1           CREATE TABLESPACE HRTS1 IN HRDB
****************************** END OF DB2 DATA ******************************


+---------------------------------------------------------------------------+
| Change "JOHNSON"."CR_HRDB" cannot be recovered now because the following  |
| changes must be recovered first.                                          |
+---------------------------------------------------------------------------+
```

Figure 490. Example of list of changes that must be recovered

- If the change can be recovered but recovering the change will cause other changes in ANALYZED status to be set to DEFINED status, a panel is displayed with the list of changes that will be set to DEFINED status.

  The following figure shows an example of the panel that might be displayed when recovering a change will cause the status of other changes to be set to DEFINED.

```
  DB2 Admin ---------------- DB2X CM - Recover Strategy ----------- Row 1 from 1
  Command ===>                                             Scroll ===> PAGE

  Commands: NEXT
  Recover strategy for change "JOHNSON"."CR_HRDEPT"
  Line commands:
   CH - Change  I - Interpret

      Rcvr
  Sel Order Owner    Name              Statement
        * *        *                 *
  --- ----- -------- ------------------ ---------------------------------------->
        0 JOHNSON  MOD_HREMP2         ALTER TABLE HREMP FOREIGN KEY RED (WORKDE
  ***************************** END OF DB2 DATA ******************************


  +-----------------------------------------------------------------------------+
  | These pending changes need to be superseded in order for the change to be   |
  | recovered. Each of these changes that are not in DEFINED status will be set |
  | to DEFINED. You must ensure the PCACT parameter in the recover job is set   |
  | to supersede (e.g. PCACT(SUPERSEDE)) to confirm the supersede action.       |
  | Note: The recover strategy is re-calculated at runtime and thus may be      |
  | different from what it is now.                                              |
  +-----------------------------------------------------------------------------+
```

Figure 491. Example of list of changes that will be set to DEFINED status

3. If a panel is displayed that shows the changes in DEFINED status and changes in ANALYZED status that will be set to DEFINED status, review the list of changes. Issue the NEXT command to proceed with recovering the change.
4. Edit and submit the generated job. When the job completes successfully, the status of the change that is recovered is set to DEFINED and the status of the recover change is set to COMPLETE.
5. Press PF3 to return to the Changes panel to verify that the status of the change is DEFINED and the status of the recover change is COMPLETE.

   Tip: If you return to the Changes panel before the submitted job completes, you can enter the REF primary command after the job completes to see the refreshed status of the change.

What to do next

If the job fails, check the job output to determine the cause of failure, make the necessary corrections, and restart the job.

Restriction: The following restrictions apply to recovering changes:
- If an ignore was specified for a change, the change cannot be recovered.
- If privileges were granted as part of the change that was recovered, the privileges are not revoked when the change is recovered. You must create a new change to revoke the privileges. Changes to revoke privileges can be made through Change Management only if they are run as immediate changes.
- If you rotate a table partition multiple times, you can only recover the most recent change.

Modifying a change

You can modify the change statements in an existing change if the change is in INITIAL, DEFINED, or ANALYZED status.

About this task

However, modifying an existing change is considered a manual intervention and is not recommended for several reasons. When you modify an existing change, DB2 Admin cannot apply virtual changes or determine whether pending changes exist. Modifying an existing change can also impact other existing changes substantially. For example, the change you are modifying might be a pending change that was applied when another change was created.

During the process of modifying a change, DB2 Admin checks only the syntax of each change statements. When you modify change statements through the Change Statements panel (ADB2C1S) panel, for example, syntax checking is completed at the time that you exit the panel. Semantic checking is done during the analyze process.

To modify the change statements in an existing change:

Procedure

1. Identify and consider the impact of the changes to dependent changes. For example, assume that want to modify a change that adds a new column to a table to change the name of the column. The change might be a prerequisite change to other changes that use that column such as another change that creates an index that includes that column.

2. Display the change to be modified by selecting option 1 on the Change Management panel, and then select option 1 on the Manage Changes panel.

3. Issue the ST line command to display the change statements in the change that you want to modify on the Change Statements panel. The following figure shows an example of the Change Statements panel:

```
DB2 Admin ------------- CM - Change Statements ------------------ Row 1 from 1
Command ===>                                               Scroll ===> CSR

Change statements for change "JOHNSON"."EMP_CH2"
Line commands:
 E - Edit  D - Delete  I - Insert  S - Show

Sel    Sequence O  Qual     Name               Statement
             * *  *         *                  *
--- ----------- -- -------- ------------------ ------------------------------->

          1 TB JOHNSON  HREMP              ALTER TABLE "JOHNSON"."HREMP"
****************************** END OF DB2 DATA ******************************
```

Figure 492. Change Statements panel (ADB2C1S)

4. Issue the E line command to change any of change statements in the change, the D line command to delete a change statement, and the I line command to insert a new change statement. When you use the E and I line commands, you are put into an ISPF edit session and can work with the SQL statement.

5. Press F3 to return to the Change Statements panel. DB2 Admin reregisters the change. A message is displayed to indicate whether the change was registered successfully. When a modified change is reregistered, pending changes or prerequisite changes are not processed.

6. Reanalyze any change that is in ANALYZED status and that is impacted by the modifications that you made to this change. Reanalyzing the impacted changes ensures the validity of the changes.

Deleting a change

You can delete certain types of changes if DB2 Admin has been configured to support the delete change line command and you have the appropriate privileges.

About this task

If the requested change is deleted and has a recover change, the recover change is also deleted.

You can delete only changes that have a type of COMPARE, FAST, CHANGE, or RECOVER.

To delete a change:

Procedure

1. Display the change to be deleted by selecting option 1 on the Change Management panel, and then select option 1 on the Manage Changes panel.
2. Issue the DEL line command against the change that you want to delete. A pop-up window is displayed to confirm your intention to delete the change.
3. Select 2 to continue with deleting the change.

Results

After a change is deleted, change no longer appears in the list of changes. The change is removed from the Change Management database, which removes any audit tracking for the change.

Promoting changes

Promoting changes allows you to move changes from one system to another because a delta changes data set is generated, which you can then import into a change on another system.

About this task

To promote a change, two versions must exist. The *starting version* represents the state of objects before any changes are made and the *ending version* represents the state of objects after the promoted changes are made. During the promote process, DB2 Admin compares the ending version with the starting version and generates a delta changes data set the contains the SQL statements that are required to bring the other system up to the same level as the system from which your promoting the changes.

To promote a change:

Procedure

1. Select option 1 on the Change Management panel, and then select option 3 on the Manage Changes panel to display the Promote panel.

   Alternatively, you can use either of the following methods to display the Promote panel:

   - If you know the ending version, specify the PR line command for the version on the Versions panel. The Promote panel will be displayed with the information for the ending version filled in.

- If you know the change and a new base version was created when the change was run, specify the PR line command for the change on the Changes panel. The Promote panel will be displayed with the information for the ending version filled in.

2. Specify the following information on the Promote panel and press Enter.
   - The starting version
   - The ending version
   - The data set name for the promote batch job
   - The data set name for the delta changes statements

   The following figure shows an example of the Promote panel:

```
DB2 Admin ---------------------- CM - Promote ------------------------- 18:33
Command ===>

Start Version (Old):
  Owner . . . . . . JOHNSON  >           (? to look up)
  Name  . . . . . . HR_VER1            > (? to look up)

 or enter a data set name that contains a Start Version:

  Data set name . .

End Version (New):
  Owner . . . . . . JOHNSON  >           (? to look up)
  Name  . . . . . . HR_VER2            > (? to look up)

Output data set names:
  Promote JOB JCL . DSN8.PROMOTE.JCL
  Delta change  . . PROMOTE.CH.HR01
```

Figure 493. Promote (ADB2CPS) panel

3. Specify the following information on the register panel and issue the CONTINUE command:
   a. Specify an owner and a name for the change. The default owner is the current SQL ID.
   b. Optionally, specify a comment for the change, an ignore for the change, and a mask for the change.

   The change will be registered as a COMPARE change.

4. Edit and submit the generated job. When the job completes successfully, the change is placed in COMPLETE status.

Results

You can now import the delta changes data set into a new change on another system, analyze the change, and run the change to bring the level of the other system up to the level of the current system.

Importing changes

You can create a change by importing SQL statements from a data set. When you import the statements, a new change is created and registered. You can import multiple delta changes as a group in one change.

About this task

When importing changes:

- You can import SQL statements (DDL) and you can import delta changes as generated by the DB2 Object Comparison Tool.
- DB2 Object Comparison Tool will create a delta change if "CHANGE" is specified for "Generate apply jobs" on panel GOC5 (this generates parameter CMDELTA for GOC2CMP).
- Importing an DB2 Object Comparison Tool change data set that is not generated as a delta change can have unwanted side effects. This cannot be checked during import.
- You can import a mix of SQL statements and delta changes (as long as they logically relate).

If the imported SQL statements affect objects for which pending changes exist, you determine whether the imported change becomes a prerequisite change for those pending changes or not.

You can create a single change by importing multiple files at the same time. Each file must be one of the types mentioned in the following list. All types can be part of the same Import.

The data sets from which you are importing the SQL statements must be either:
- The delta changes data set that was generated when changes were promoted with Change Management from another system. Thus, you can import the changes that were promoted from another system that uses Change Management.
- The delta changes data set that DB2 Object Comparison Tool generated when objects were compared. DB2 Object Comparison Tool uses the worklist name for this data set (*qualifier1.worklist_name*.CHG). The worklist name will be generated by specifying CHANGE in the "Generate apply jobs" field on panel GOC5. You can specify a data set name and optionally specify a member name if the data set is partitioned.
- A data set that contains SQL statements that meets these requirements:
  - A fixed-block sequential data set (`RECFM=Fx,LRECL=80`)
  - A member of a partitioned data set with a logical record length of 80 (`RECFM=Fx,LRECL=80`)

During the import process, the syntax of each change statements in imported SQL statements is checked. However, semantic checking is done during the analyze process.

If you are importing a delta changes data set, the data set must represent one generated delta changes file. Concatenating or merging multiple data sets into one can cause unpredictable results because statements are reordered during the import process.

When you import SQL statements into a change, the subsystem being used for the IMPORT must support the SQL statements that you are importing.

To import a change:

Procedure

1. Select option 1 on the Change Management panel to display the Manage Changes panel.
2. Select option 4 to import changes.

3. Specify the name of the data set that contains the SQL statements. This panel is re-displayed after each entry so that you can enter more input data sets. Thus, you can generate a list of input dataset names that will be processed in the specified sequence. The following figure shows the Import Changes panel:

```
ADB2C14 n ------------------ CM - Import Changes ------------------ 08:05
Command ===

Commands : CONTINUE RESET                              DB2 System: DSN8

Input data set information:
  Data set name .
    Member  . . .         (member name or pattern if partitioned)
Line commands :
M - Move   A - After  B – Browse  D - Delete

Select Seq Data set name                                           Oper.
------ --- -------------------------------------------------------    ------

         1 USER01.PROD.CHANGES(FEB08001)
         2 USER01.PROD.CHANGES(FEB08002)
         3 USER01.PROD.CHANGES(FEB08003)
         4 USER01.PROD.CHANGES(FEB08010)
         5 USER01.PROD.CHANGES(FEB08011)
         6 USER01.PROD.CHANGES(FEB08071)
         7 USER01.PROD.CHANGES(FEB08072)
         8 USER01.PROD.CHANGES(FEB08073)
         9 USER01.PROD.CHANGEXX
        10 USER01.PROD.CHANGES(XXCHGA)
        11 USER01.PROD.CHANGES(XXCHGB)
****************************** END OF DB2 DATA ******************************
```

Figure 494. Import Changes panel (ADB2C14)

If the input dataset is a PDS, you must specify a member name or a member pattern (as defined by ISPF) . If a member pattern is specified, all members that fit the pattern will be added to the list of data sets to import in member name sequence. If you want a different sequence, you can use line commands to move entries in the list

To process the import, issue the CONTINUE command. To clear the list of data sets, issue RESET.Importing multiple data sets into a single change should be carefully planned. Import cannot check whether the changes in the specified sequence will logically work as desired. The input changes will be imported into the change individually in the sequence they are specified, and you must ensure that any change in the list logically has all preceding changes as prerequisites.

4. Importing a change is a two-phase process in which DB2 Admin determines if there are any pending changes for the objects and then registers the imported change. The processing modes are:
   TSO   Perform the processing in the foreground (TSO)
   Batch   Perform the processing in background (batch)

   The following figure shows the Import Changes - Select process modes panel:

```
ADB2C14M  ------- CM Import changes - Select process modes --14:28


 Specify how to continue Import :

 Prereq resolution mode  . BATCH     (TSO/Batch)
 Execution mode  . . . . . BATCH     (TSO/Batch)
```

Figure 495. Import Changes - Select process modes panel (ADB2C14M)

5. If you specify TSO for both prerequisite checking and change registration, complete the following steps:

   a. Fill in the fields of the Register panel, and issue the CONTINUE command. Specify the following information:

      - Specify an owner and a name for the change. The default owner is the current SQL ID.

      - Optionally, specify a comment for the change, an ignore for the change, and a mask for the change.

   b. If the changes in the data set affect objects that have pending changes, specify the action to take on the Import Pending panel and press Enter. The following actions are possible:

      Prereq Make the pending changes for the objects prerequisite changes for the imported change

      Supersede
      Make the imported change a prerequisite change for the pending changes

      Cancel
      Cancel importing the change

      Display
      Display the changes that are pending

      Ignore Ignore pending changes. Pending changes are not set to DEFINED status. If you choose the Ignore option, you should ensure that pending changes do not conflict with current changes before you register any changes to your objects. You should use run-time analyze when running the change to identify any conflicting changes.

   c. When the Import Changes panel is re-displayed, verify the message that indicates whether the change was registered successfully. The change is put in DEFINED status. If you selected the Ignore option, pending changes are not put in DEFINED status.

6. If you specify TSO for prerequisite checking and batch for change registration, complete the following steps:

   a. Specify a data set name to contain the (delta) change statements and press Enter.

   b. Fill in the fields on the Register panel, and issue the CONTINUE command. Specify the following information:

      - Specify an owner and a name for the change. The default owner is the current SQL ID.

      - Optionally, specify a comment for the change, an ignore for the change, and a mask for the change.

   c. If the changes in the data set affect objects that have pending changes, specify the action to take on the Import Pending panel and press Enter. The possible actions are:

      Prereq Make the pending changes for the objects prerequisite changes for the imported change.

      Supersede
      Make the imported change a prerequisite change for the pending changes

      Cancel
      Cancel importing the change

      Display
      Display the changes that are pending

      Ignore Ignore pending changes. Pending changes are not set to DEFINED

status. If you choose the Ignore option, you should ensure that
pending changes do not conflict with current changes before you
register any changes to your objects. You should use run-time
analyze when running the change to identify any conflicting
changes.

d. Review the job to register the change and submit the JCL. When the job
completes successfully, the change is registered and put in DEFINED status.
If you selected the Ignore option, pending changes are not put in DEFINED
status.

7. If you use batch mode for resolving prerequisite changes, you must use batch
mode for registering the change. If you specify batch for both prerequisite
checking and change registration, complete the following steps:

a. Specify the action to take if there are pending prerequisite changes for the
objects that the imported change affects. The options on the Import a
Change - Action for Pending Changes panel are:

Prereq Make the pending changes for the objects prerequisite changes for
the imported change.

Supersede
Make the imported change a prerequisite change for the pending
changes

Cancel
Do not import the changes if there are pending changes

Ignore Ignore pending changes. Pending changes are not set to DEFINED
status. If you choose the Ignore option, you should ensure that
pending changes do not conflict with current changes before you
register any changes to your objects. You should use run-time
analyze when running the change to identify any conflicting
changes.

Recommendation: Specify Cancel to avoid registering the changes if there
are pending changes. You can review the batch output, which will list the
pending changes, decide whether to keep them as prerequisite changes or
supersede them, and then import the change again specifying either Prereq
or Supersede.

b. Fill in the fields of the register panel, and issue the CONTINUE command.
Specify the following information:

• Specify an owner and a name for the change. The default owner is the
current SQL ID.

• Optionally, specify a comment for the change, an ignore for the change,
and a mask for the change.

c. Review the job and submit the JCL.

Results

You can now display your imported change on the Changes panel, analyze the
change, and then run it.

Masks

A *mask* (also called *translation mask*) provides the ability to cause context-sensitive
global changes to naming conventions and to overwrite the current values of
certain table space and index space attributes when you use various functions of
DB2 Admin and DB2 Object Comparison Tool.

You can define and manage masks by using the Change Management panels. Masks that are specified when you import changes through Change Management must be defined in the Change Management database, where the masks are stored in a table. Masks that are specified on panels for reverse engineering the catalog, cloning WSLs, migrating objects, or explicitly performing comparisons by using DB2 Object Comparison Tool can be defined in the Change Management database or in a data set.

Tip: Consider managing all your masks through Change Management. The masks are easy to track and recover because they are stored in the Change Management database.

The Manage Masks panel, as shown in the following figure, is the main menu for working with masks.

```
DB2 Admin ------------------- CM - Manage Masks   ---------------------- 18:03
Option ===>


   1 - Display masks                                   DB2 System: DB2X
   2 - Create a mask                                   DB2 SQL ID: ISTJE







Enter display selection criteria (Using a LIKE operator, criteria not saved):
Name . . . . . . .                  > Created by . .         >
Owner  . . . . . .          >         Altered by . .         >
Created within . .                    Mask ID  . . .
Altered within . .
```

Figure 496. Manage Masks panel (ADB2C2)

Displaying the masks

You can display the masks that are stored in the Change Management database.

About this task

To display the masks:

Procedure

1. Select option 2 on the Change Management (CM) panel to display the Manage Masks panel.
2. Optional: Specify the search criteria to filter or limit the masks that are displayed.
3. Select option 1 on the Manage Masks panel to display the Masks panel, shown in the following figure:

```
DB2 Admin ------------------ CM - Masks ------------------ Row 1 to 10 of 10
Command ===>                                          Scroll ===> PAGE

Line commands:
 U - Update  DEL - Delete  INS - Insert  ML - Mask lines  CH - Changes
 E - Edit  I - Details on mask

Sel          ID Owner    Name              Comment
              * *        *                 *
--- ----------- -------- ----------------- --------------------------------
             14 JOHNSON  DEVHRMASK         MASK FOR HR APPLICATION
             16 JOHNSON  TSTBANKMASK       MASK FOR BANKING APPLICATION
             21 TONELLO  MYFIRSTMASK
             41 MYID     MYMASK            MY MASK IN CM
             42 MYID     MYMASK1           ANOTHER NEW MASK
             43 MYID     MYMASK2
             45 MYID     MYMASK3
             47 MYID     MYMASK4
             61 LOSER    LOSER             MASK FOR LOSERS
****************************** END OF DB2 DATA ******************************
```

Figure 497. Masks panel (ADB2C31)

Results

You can issue a variety of line commands for each mask that is displayed on the
Masks panel. Commands are available to do the following tasks:
- See the definition of the mask and modify it
- View details about who created the mask and when and who altered it last
- See which changes use the mask
- Insert, delete, or update a mask

Masks that have been created in an explicitly named data set outside of Change
Management are not displayed because they are not stored in the Change
Management database. You might have created masks that you use when
performing comparisons using DB2 Object Comparison Tool or other functions in
DB2 Admin (such as reverse engineering, migrating DB2 data, or cloning work
statement lists) in a data set. When you are prompted to specify the mask to use,
you have the option of using masks that are either in data sets or in the Change
Management database.

Creating a mask

You can create a mask that is stored in the Change Management database.

About this task

To create a mask:

Procedure

1. Select option 2 on the Change Management (CM) panel to display the Manage
   Masks panel.
2. Select option 2 on the Manage Masks panel to display the Insert Mask panel.
3. Specify an owner and a name for the mask, and optionally enter a comment for
   the mask. Press Enter.
4. Press F3 to return to the Manage Masks panel.
5. Select option 1 to display the masks on the Masks Panel.
6. Issue the ML line command for the mask you just created to add the mask line
   definitions for the mask. For each mask line that you add, specify:

- The type of object for the mask in the Type field. For example, TBNAME specifies a mask for tables.
- The input mask (the pattern of the string that you want to translate) in the From field.
- The output mask (the string to which you want to translate) in the To field.

For example, to define a mask that translates any table name that starts with DEV to a name that starts with TST and a column name from CELLNO to MOBILENO, enter the values that are shown in the following figure:

```
DB2 Admin ----------------- CM - Mask Lines -------------------- Row 1 from 2
Command ===>                                             Scroll ===> PAGE

Mask lines for mask "MYID"."MYMASK2"
Commands: SAVE
Line commands:
  I - Insert  D - Delete  R - Repeat  M - Move  A - After  B - Before

Sel    Sequence Type      From                 To                   Oper.  T
                * *       *                    *                    *      *
--- ----------- --------- -------------------> -------------------> ------ -

*             1 TBNAME    TB_TEST              TB_PROD              UPDATE
*             2 COLNAME   CELLNO               MOBILENO             UPDATE
*             3 SINGLECH  _                    +
*             4 ALNAME    ALS+_TEST            ALS+_PROD
**************************** END OF DB2 DATA ******************************
```

Figure 498. Mask Lines panel (ADB2C2L)

The hierarchy of mask types is the same as it is when you define and edit a mask data set outside of Change Management. See the online help to review the hierarchy.

You can use the I, D, and R line commands to quickly insert, delete, and repeat mask lines when you define a mask. You can also use the A and B line commands with the M line command to quickly move the mask lines around in the definition. The order of the mask lines in the definition is important because the first mask that matches is used and the name is translated to the second value. You should put the most specific translation masks at the beginning and the more general ones at the end.

7. Issue the SAVE primary command to save the definition of the mask.
8. Press F3 to return to the Manage Masks panel.

Creating and storing a new mask in the Change Management database
About this task

If you are using DB2 Object Comparison Tool or other functions in DB2 Admin (such as reverse engineering, migrating DB2 data, or cloning work statement lists) and specify to use masking and you need to define a new mask, you can specify that the mask that be created and stored in the Change Management database instead of a data set if Change Management is enabled. When you are prompted to specify the masks on either the Specify Compare Masks panel or the Specify Masks panel, complete the following steps:

Procedure

1. As shown in the following figure, specify an owner and a name for the mask, do not specify a data set name, specify YES in the **Edit Mask** field, and press Enter.

```
   Compare -------------------- Specify Compare Masks -------------------------
   Option ===>


   Mask Table Entry:
     Owner . .  MYID    >           (? to look up)
     Name  . .  MYMASK           >  (? to look up)
   Data Set:
     Mask DSN  . .
   Options:
     Edit Mask . .  YES (Yes/No)
```

Figure 499. Specify Compare Masks panel (GOC3)

2. Verify the owner and name of the mask on the owner. Optionally, enter a comment for the mask. Press Enter. A message is displayed that indicates that the mask was inserted.

3. Press F3 to display the Mask Lines panel to define the entries in the mask. For each mask line that you add, specify:
   - The type of object for the mask in the Type field. For example, TBNAME specifies a mask for tables.
   - The input mask (the pattern of the string that you want to translate) in the From field.
   - The output mask (the string to which you want to translate) in the To field.

4. Issue the SAVE primary command to save the definition of the mask.

Editing a mask

You can change the definition of a mask.

About this task

To edit a mask that is stored in the Change Management database:

Procedure

1. Select option 2 on the Change Management (CM) panel to display the Manage Masks panel.

2. Select option 1 on the Manage Masks panel to display the masks on the Masks panel.

3. Issue either the ML line command or the E line command for the mask you want to edit.
   - When you use the ML line command, you use the Mask Lines panel to add, delete, and change the definitions for your mask. Each line in the file defines a mask type. You can use the I, D, and R line commands to quickly insert, delete, and repeat mask lines when you edit the mask. You can also use the A and B line commands with the M line command to quickly move the mask lines around in the definition. Issue the SAVE primary command to save your changes. Press PF3 to return to the Masks panel.
   - When you use the E line command, you use ISPF edit to edit the mask data set that contains the mask definition. Press PF3 to save your changes and return to Masks panel.

Deleting a mask

You can delete a mask that is stored in the Change Management database.

About this task

To delete a mask:

Procedure

1. Select option 2 on the Change Management (CM) panel to display the Manage Masks panel.
2. Select option 1 on the Manage Masks panel to display the Masks panel.
3. Issue the DEL line command for the mask that you want to delete.

Ignores

An *ignore* provides the ability to specify that certain fields in the DB2 catalog records are to be ignored when objects are compared.

Objects are compared when you analyze a change or you explicitly use DB2 Object Comparison Tool to generate a compare job.

Overview of ignores

You can define and manage ignores by using the Change Management panels.

Ignores that are specified when analyzing a change must be defined in the Change Management database, where the ignore is stored in a table. Ignores that you specify when you explicitly use DB2 Object Comparison Tool to generate a compare job can be either in the Change Management database or in a data set.

The purpose of ignoring fields is to:

- Avoid comparisons that are meaningless

  Timestamps and statistical information are examples of this type of information. These types of ignore fields are called *system ignores* and are automatically included by default.

- Protect specified fields against updates

  Examples of fields that you might want to ignore are fields that contain space information because production tables and indexes are often larger than the corresponding test tables and indexes. You might also want to ignore fields that contain buffer pool names because a broader set of pools might be implemented in the production system.

No field in a DB2 catalog record for which an ignore is specified is compared. If you must re-create an object because of other changes, values for ignored fields are taken from the target version. All other fields have values taken from the source version.

Some catalog fields are automatically ignored, such as statistics, dates, and internal identifiers. As mentioned previously, these fields are called *system ignores*.

Use caution when specifying ignore fields. If possible, use the generic specifications, which provide for some common sets of fields that are often intentionally different on source and target systems.

Because many fields in the DB2 catalog records are interdependent, when one field is ignored, the value in another field might be invalid if that field is not ignored also, for example, the TYPE fields for tables and table spaces. If TYPE is ignored for table spaces, a table space could keep the LARGE (TYPE) attribute. If the

compare source is a segmented table space, the resulting set of attributes will be invalid if the SEGSIZE field is not ignored also.

Another type of dependency is between the SQTY and SECQTYI fields on SYSTABLEPART and SYSINDEXPART that are updated by DB2. If secondary quantity is to be ignored, specify both fields or use the generic SPACE specification.

Tip: Consider managing all your ignores through Change Management. The ignores are easy to track and recover because they are stored in the Change Management database.

Ignore fields

Only certain fields in certain DB2 catalog tables can be ignored.

The following table shows the DB2 catalog tables and the ignore fields that you can specify.

Table 22. The DB2 catalog table ignore fields

DB2 catalog table	Ignore fields
SYSCHECKS	CREATOR, CHECKCONDITION
SYSCOLUMNS	COLTYPE, LENGTH, SCALE, NULLS, REMARKS, DEFAULT, KEYSEQ, FOREIGNKEY, FLDPROC, LABEL, DEFAULTVALUE, LENGTH2, TYPESCHEMA, TYPENAME, STATS_FORMAT, PARTKEY_COLSEQ, PARTKEY_ORDERING, ALTEREDTS
SYSDATABASE	CREATOR, STGROUP, BPOOL, ROSHARE, TYPE, GROUP_MEMBER, ENCODING_SCHEME, SBCS_CCSID, DBCS_CCSID, MIXED_CCSID, INDEXBP
SYSDATATYPES	OWNER, SOURCESCHEMA, SOURCETYPE, METATYPE, LENGTH, SCALE, SUBTYPE, ENCODING_SCHEME, REMARKS
SYSFIELDS	FLDPROC, WORKAREA, EXITPARML, PARMLIST, EXITPARM
SYSINDEXES	UNIQUERULE, CLUSTERING, BPOOL, PGSIZE, ERASERULE, DSETPASS, CLOSERULE, INDEXTYPE, PIECESIZE, COPY, SPACEF, REMARKS, PADDED, VERSION, OLDEST_VERSION, CURRENT_VERSION, RELCREATED, AVGKEYLEN
SYSINDEXPART	PARTITION, PQTY, SQTY, STORTYPE, STORNAME, VCATNAME, LIMITKEY, FREEPAGE, PCTFREE, INDEXTYPE, GBPCACHE, SECQTYI, SPACEF, DSNUM, EXTENTS, PSEUDO_DEL_ENTRIES, LEAFNEAR, LEAFFAR
SYSSYSKEYS	COLSEQ, ORDERING
SYSPARMS	OWNER, SPECIFICNAME, CAST_FUNCTION, PARMNAME, ROWTYPE, ORDINAL, TYPESCHEMA, TYPENAME, LOCATOR, TABLE, TABLE_COLNO, LENGTH, SCALE, SUBTYPE, CCSID, ENCODING_SCHEME
SYSRELS	RELNAME, DELETERULE, IXOWNER, IXNAME, ENFORCED, CHECKEXISTINGDATA
SYSROUTINES	OWNER, CREATEDBY, SPECIFICNAME, RETURN_TYPE, ORIGIN, FUNCTION_TYPE, PARM_COUNT, LANGUAGE, COLLID, SOURCESCHEMA, SOURCESPECIFIC, DETERMINISTIC, EXTERNAL_ACTION, NULL_CALL, CAST_FUNCTION, SCRATCHPAD, SCRATCHPAD_LENGTH, FINAL_CALL

Table 22. The DB2 catalog table ignore fields (continued)

DB2 catalog table	Ignore fields
SYSSEQUENCES	OWNER, SEQTYPE, INCREMENT, START, MAXVALUE, MINVALUE, CYCLE, CACHE, ORDER, REMARKS, PRECISION, RESTARTWITH
SYSTABLEPART	IXNAME, IXCREATOR, PQTY, SQTY, STORTYPE, STORNAME, VCATNAME, LIMITKEY, FREEPAGE, PCTFREE, COMPRESS, GBPCACHE, TRACKMOD, SECQTYI, SPACEF, DSNUM, EXTENTS, LOGICAL_PART
SYSTABLES	TYPE, DBNAME, TSNAME, EDPROC, VALPROC, CLUSTERTYPE, REMARKS, KEYCOLUMNS, STATUS, LABEL, AUDITING, CREATEDBY, LOCATION, TBCREATOR, TBNAME, DATACAPTURE, CHECKS, ENCODING_SCHEME
SYSTABLESPACES	CREATOR, BPOOL, PARTITIONS, LOCKRULE, PGSIZE, ERASERULE, STATUS, IMPLICIT, DSETPASS, CLOSERULE, SEGSIZE, LOCKMAX, TYPE, ENCODING_SCHEME, SBCS_CCSID, DBCS_CCSID, MIXED_CCSID, MAXROWS
SYSTRIGGER	OWNER, TRIGTIME, TRIGEVENT, GRANULARITY, TEXT, REMARKS, TRIGNAME
SYSVIEWS	CHECK, TEXT, PATHSCHEMAS, RELCREATED, TYPE, REFRESH, ENABLE, MAINTENANCE, REFRESH_TIME, ISOLATION, SIGNATURE, APP_ENCODING_CCSID

When you specify ignore fields for SYSCOLUMNS, consider the following information:

- The fields COLTYPE, LENGTH, SCALE, DEFAULT, and DEFAULTVALUE are all part of the column type definition. The NULLS field is also related because in some cases it is part of the default specification.
- The DEFAULT field can have a relationship to a SYSSEQUENCES row. Ignoring the DEFAULT field can cause the SYSSEQUENCES row to be included or excluded, depending on the value of the DEFAULT field in the target SYSCOLUMNS row. However, to ignore fields in the SYSSEQUENCES row, you must explicitly select them.
- The FOREIGNKEY field specifies the subtype of a character type column. Ignoring the FOREIGNKEY field not only removes the check for SBCS and MIXED data, but also the FOR BIT DATA specification (that is, CCSID conversions will occur, if applicable).
- The FLDPROC field can have a relationship to a SYSFIELDS catalog row. Ignoring the FLDPROC field can cause the SYSFIELDS row to be included or excluded, depending on the value of FLDPROC in the target SYSCOLUMNS row. However, to ignore fields in the SYSFIELDS row, you must explicitly select them.

Important: Be careful when you choose to ignore some, but not all, of the fields that are part of a column definition. Otherwise, it is possible that inconsistent attributes and, subsequently, invalid DDL will result.

Generic ignore fields

Generic ignore field specifications provide a shortcut for ignoring all buffer pools, allocated space information, and information about how data is stored and partitioned. The generic ignore specifications are:
- BUFFERPOOL
- SPACE
- STORAGE

- PARTITIONING

Specifying a generic ignore specification has the same effect as specifying the ignore fields individually. The following table shows which catalog fields are ignored when the generic ignore specification is selected.

Table 23. Generic ignore specifications

Generic ignore specification	DB2 catalog table	Ignore fields
BUFFERPOOL	SYSDATABASE	BPOOL, INDEXBP
	SYSINDEXES	BPOOL
	SYSTABLESPACE	BPOOL
SPACE	SYSINDEXPART	PQTY, SQTY, FREEPAGE, PCTFREE, SECQTYI
	SYSTABLEPART	PQTY, SQTY, FREEPAGE, PCTFREE, SECQTYI
	SYSTABLESPACE	MAXROWS
STORAGE	SYSDATABASE	STGROUP
	SYSINDEXPART	STORTYPE, STORNAME, VCATNAME
	SYSTABLEPART	STORTYPE, STORNAME, VCATNAME
	SYSSTOGROUP	VCATNAME
PARTITIONING	SYSINDEXPART	PARTITION
	SYSTABLEPART	PARTITION
	SYSTABLESPACE	PARTITIONS
	SYSINDEXPART	LIMITKEY
	SYSTABLEPART	LIMITKEY
	SYSTABLEPART	LIMITKEY_INTERNAL
	SYSTABLEPART	LOGICAL_PART
	SYSTABLES	PARTKEYCOLNUM
	SYSCOLUMNS	PARTKEY_COLSEQ
	SYSCOLUMNS	PARTKEY_ORDERING
	SYSAUXRELS	PARTITION

The Manage Ignores panel

The Manage Ignores panel is the main menu for working with ignores.

The following figure shows the Manage Ignores panel:

```
DB2 Admin ------------------ CM - Manage Ignores ---------------------- 20:10
Option ===>


  1 - Display ignores                                DB2 System: DB2X
  2 - Create an ignore                               DB2 SQL ID: ISTJE









Enter display selection criteria (Using a LIKE operator, criteria not saved):
Name . . . . . . .               > Created by . .        >
Owner  . . . . . .        >          Altered by . .        >
Created within . .                   Ignore ID  . .
Altered within . .
```

Figure 500. Manage Ignores panel (ADB2C3)

From the Manage Ignores panel, you can display the existing ignores to work with them or create a new ignore.

Displaying the ignores

You can display the ignores that are stored in the Change Management database.

About this task

To display the ignores:

Procedure

1. Select option 3 on the Change Management (CM) panel to display the Manage Ignores panel.
2. Optional: Use the fields at bottom of the panel to enter the search criteria to filter or limit the ignores that are displayed.
3. Select option 1 on the Manage Ignores panel to display the Ignores panel, as shown in the following figure:

```
DB2 Admin ------------------ CM - Ignores ------------------  Row 1 to 8 of 33
Command ===>                                            Scroll ===> PAGE

Line commands:
 U - Update  DEL - Delete  INS - Insert  IL - Ignore lines  CH - Changes
 I - Details on ignore

Sel        ID Owner    Name             Comment
           * *         *                *
--- ----------- -------> ----------------> --------------------------------
            1 J148286  HRIGNORE_BUFFER  IGNORE BUFFERPOOL FOR HR
            2 JOHNSON  EMPIGNORE1       TEST IGNORE1 FOR EMP TABLE
           21 JOHNSON  EMPIGNORE2       TEST IGNORE2 FOR EMP TABLE
           41 J148286  DEVTS            IGNORE PARTITIONING
           47 J148286  HRIGNORE_VCAT    IGNORE VCAT FOR HR
           48 J148286  DEVSYS1
           49 KINCAID  TESTSYS1         IGNORE SPACE
           50 KINCAID  TESTSYS2         IGNORE SPACE
```

Figure 501. Ignores panel (ADB2C31)

Results

You can issue a variety of line commands for each ignore that is displayed on the Ignores panel. Commands are available to:
- See the definition of the ignore and modify it
- View details about who created the ignore and when and who altered it last
- See which changes use the ignore
- Insert, delete, or update a ignore

Ignores that have been created in an explicitly named data set outside of Change Management on the Specify Ignore Fields panel (GOC4) in DB2 Object Comparison Tool are not displayed because they are not stored in the Change Management database.

Creating an ignore

You can create an ignore that is stored in the Change Management database.

About this task

To create an ignore:

Procedure

1. Select option 3 on the Change Management (CM) panel to display the Manage Ignores panel.
2. Select option 2 on the Manage Ignores panel to display the Insert Ignore panel.
3. Specify an owner and a name for the ignore, and optionally enter a comment for the ignore.
4. Press F3 to return to the Manage Ignores panel.
5. Select option 1 to display the ignores on the Ignores panel.
6. Issue the IL line command for the ignore that you just created to define the ignore fields. The Specify Ignore Fields: Objects panel is displayed, as shown in the following figure. The panel shows the DB2 catalog tables for which you can define ignore fields.

```
 ---------------- Specify Ignore Fields: Objects ----------- Row 1 to 18 of 18
Command ===>                                                 Scroll ===> PAGE

Valid line commands are:
 U - Update Ignore Fields

Select Object         Ignore Fields
       *              *
------ --------------  -------------------------------------------------------
       GENERIC         None
       SYSCHECKS       None
       SYSCOLUMNS      None
       SYSDATABASE     None
       SYSDATATYPES    None
       SYSFIELDS       None
       SYSINDEXES      None
       SYSINDEXPART    None
       SYSKEYS         None
       SYSPARMS        None
       SYSRELS         None
       SYSROUTINES     None
       SYSSEQUENCES    None
       SYSTABLEPART    None
       SYSTABLES       None
       SYSTABLESPACE   None
       SYSTRIGGERS     None
       SYSVIEWS        None
```

Figure 502. Specify Ignore Fields: Objects panel (GOCCI)

7. For each table, use the U line command to display the catalog field columns that can be ignored.
8. On the Select Ignore Fields panel for the table, use the U and S line commands to select or de-select a particular field to be ignored.
9. Press F3 to return to the list of DB2 catalog tables (the Specify Ignore Fields: Objects panel). Pressing F3 repeatedly returns you through the panels to the main menu.

Storing an ignore in the Change Management database
About this task

If you are using DB2 Object Comparison Tool and choose option 4 on the DB2 Object Comparison Tool Menu to specify the fields to ignore, you can specify that the ignore that is created be stored in the Change Management database instead of a data set. To have the ignore stored in the Change Management database, complete the following steps:

Procedure

1. Select option 4 on the DB2 Object Comparison Tool Menu to display the Specify Compare Ignore Fields panel.
2. As shown in the following figure, specify an owner and a name for the ignore, do not specify a data set name, and specify YES in the **Edit Ignores** field.

```
Compare ---------------- Specify Compare Ignore Fields ---------------------
Option ===>



Ignore Table Entry:

  Owner . . MYID     >             (? to look up)
  Name  . . MYIGNORE          >    (? to look up)
Data Set:
  Data Set Name . .
Options:
  Edit Ignores  . . YES (Yes/No)
```

Figure 503. Specify Compare Ignore Fields panel (GOC4)

3. For each table that is displayed on the Specify Ignore Fields: Objects panel, use the U line command to display the catalog field columns that can be ignored.
4. On the Select Ignore Fields panel for the table, use the U and S line commands to select or de-select a particular field to be ignored.
5. Press F3 to return to the list of DB2 catalog tables (the Specify Ignore Fields: Objects panel). Pressing F3 again returns you to the DB2 Object Comparison Tool Menu panel.

Editing an ignore

You can add and delete fields from an existing ignore.

About this task

To edit an ignore that is stored in the Change Management database:

Procedure

1. Select option 3 on the Change Management (CM) panel to display the Manage Ignores panel.
2. Select option 1 on the Manage Ignores panel to display the Ignores panel.
3. Issue the IL line command for the ignore that you want to edit. A list of DB2 catalog tables is displayed, and the columns that are currently selected as ignore fields are shown on the Specify Ignore Fields: Objects panel. In the example that is shown in the following figure, the CREATOR, STGROUP, BPOOL, and INDEXPB fields in SYSDATABASE and BPOOL fields in the SYSINDEXES and SYSTABLESPACES tables are to be ignored:

```
   ---------------- Specify Ignore Fields: Objects ----------- Row 1 to 18 of 18
   Command ===>                                                 Scroll ===> PAGE

   Valid line commands are:
    U - Update Ignore Fields

   Select Object            Ignore Fields
          *                 *
   ------ ---------------   -------------------------------------------------------
          GENERIC           None
          SYSCHECKS         None
          SYSCOLUMNS        None
          SYSDATABASE       STGROUP,BPOOL,INDEXBP
          SYSDATATYPES      None
          SYSFIELDS         None
          SYSINDEXES        BPOOL
          SYSINDEXPART      None
          SYSKEYS           None
          SYSPARMS          None
          SYSRELS           None
          SYSROUTINES       None
          SYSSEQUENCES      None
          SYSTABLEPART      None
          SYSTABLES         None
          SYSTABLESPACE     BPOOL
          SYSTRIGGERS       None
          SYSVIEWS          None
```

Figure 504. Example of the definition of an ignore

4. Use the U line command to display the ignore fields for a particular DB2 catalog table.
5. On the Select Ignore Fields panel for the table, use the U and S line commands to select or de-select a particular field to be ignored.
6. Press F3 to return to the list of DB2 catalog tables (the Specify Ignore Fields: Objects panel). Pressing F3 repeatedly returns you through the panels to the main menu.

Deleting an ignore

You can delete an ignore that is stored in the Change Management database.

About this task

To delete an ignore:

Procedure

1. Select option 3 on the Change Management (CM) panel to display the Manage Ignores panel.
2. Select option 1 on the Manage Ignores panel to display the Ignores panel.
3. Issue the DEL line command for the ignore that you want to delete.

Managing ignore specifications

You use DB2 Admin Tools to specify object types that you want ignored during the compare process.

Procedure

1. From the DB2 Admin Main Menu, specify option CM. The Change Management (CM) (ADB2C) panel is displayed.

2. Select option 8 - Manage ignore changes specifications. The Manage Ignore Changes Specifications (ADBPC8) panel is displayed.

```
 ADBPC8 in ----------  Manage Ignore Changes Specifications -------------------- 14:49
Option ===>


   1 - Display ignore changes specifications          DB2 System: DSNA
                                                       DB2 SQL ID: OWN1


Enter display selection criteria (Using a LIKE operator, criteria not saved):
Owner  . . . .   OWN1     >                       Created by . .         >
Name . . . . .                            > Altered by . .         >
Created within                            Exclude ID  . . .
Altered within
Eligible for auto-delete:
  Within . . .
  Next . . . .
```

Figure 505. Manage Ignore Changes Specifications panel (ADBPC8)

3. Specify the owner name and name for the ignore changes specification.
4. Optional: You can refine a search for ignore changes specifications, by using search criteria fields.
5. Select Option 1 - Display ignore changes specifications. The Ignore Changes Specifications (ADBPC81) panel is displayed.

```
 ADBPC81 n ------------ Ignore Changes Specifications ------------- Row 1 to 33 of 33
Command ===>                                            Scroll ===> CSR

Line commands:
 U - Update  DEL - Delete  ICL - Ignored Changes List
 I - Details on ignore specification

                                          Eligible for
Sel Owner    Name                         auto-delete  Comment


--- -------- ------------------------------> ----------  -------------------->
    OWN1     ICSPEC01                        2012-12-31
    OWN1     ICSPEC02
```

Figure 506. Ignore Changes Specifications panel (ADBPC81)

From the Ignore Changes Specifications (ADBPC81) panel, you can use line commands to view more detail, modify, or delete ignore changes specifications.

To modify the contents of the ignore change specification, you must work in DB2 Object Comparison Tool and select the option MR - Managed saved compare results.

Creating or managing exclude specifications

You manage lists of objects that are excluded from compare input and output processes by maintaining exclude specifications. You use DB2 Admin Tools to specify objects that you want to exclude from the compare process.

Procedure

1. From the DB2 Admin Main Menu, specify option CM. The Change Management (CM) (ADB2C) panel is displayed.
2. Select option 7 - Manage exclude specifications. The CM - Manage Exclude Specifications (ADBPC7) panel is displayed.

```
ADBPC7 in ------------  CM - Manage Exclude Specifications ------------- 10:38 .
.  Option ===>                                                                    .
.                                                                                 .
.                                                                                 .
.     1 - Display exclude specifications              DB2 System: DB2X            .
.     2 - Create an exclude specification             DB2 SQL ID: JSMITH          .
.                                                                                 .
.                                                                                 .
.  Enter display selection criteria.  Settings: LIKE operator;  Criteria not saved .
.  Owner  . . . .              >                  Created by . .            >     .
.  Name . . . . .                               > Altered by . .            >     .
.  Created within                                 Exclude ID . .                  .
.  Altered within                                                                 .
.  Eligible for delete:                                                           .
.    Within . . .                                                                 .
.    Next . . . .                                                                 .
```

Figure 507. Manage Exclude Specifications panel (ADBPC7)

3. Select an option to view an existing specification or create a new specification.

Option	Description
Edit an existing exclude specification	1. Specify Owner name or specification name. You can enter **?** to look up a name from a list.
	2. Select Option 1 - Display exclude specification.
	3. In the Exclude Specifications (ADBPC71) panel, enter the ESL line command next to a listed specification.
	If you select the ESL line command, the CM - Exclude Objects (ADBPC7L) is displayed in which you can view and edit a list of objects that are specified to be excluded in the selected exclude specification.
	4. Exit and return to the CM - Manage Exclude Specifications (ADBPC7) panel.
Create a new exclude specification	1. Select Option 2 - Create an exclude specification.
	2. In the Create Exclude Specifications (ADBPC22) panel, you specify owner name and specification name. You also can specify an **Eligible for auto-delete** value.
	3. Press Enter and in the CM - Exclude Objects (ADBPC7L) panel, insert lines and enter object names and other information.
	4. Exit and return to the CM - Manage Exclude Specifications (ADBPC7) panel.

Versions

A *version* is a snapshot of the definitions of a set of objects at a point in time.

The object definitions typically represent an application or application area.

Versions enable you to track the changes to a set of objects, restore objects to a previous version if you need to fall back, and promote changes from one system to another.

Versions can be created in one of three ways:
- When using Change Management, you can define a version scope (the objects to be included in a version) and then use the GV line command on the Version Scopes (ADB2C42) to generate a version based on that scope.
- When you run a change using Change Management, you can specify to have a version of the objects generated after the changes have been applied.
- When you use DB2 Object Comparison Tool, you can have versions of the source and target objects generated. When Change Management is enabled, you have the option of storing versions in data sets or in the Change Management database.

   Tip: Consider storing all of your versions in the Change Management database, which makes them easier to track, access, and recover.

When you promote a set of changes from one system to another, you need two versions. The *starting version* represents the state of objects before any changes are made and the *ending version* represents the state of objects after the promoted changes are made. During the promote process, DB2 Admin compares the ending version with the starting version and generates a delta changes data set the contains the SQL statements that are required to bring the other system up to the same level as the system from which your promoting the changes. You can then import the delta changes data set into a new change on the system to which you are promoting the changes, analyze the change, and run them.

When you implement them carefully, you can also use versions as the base version for subsequent changes to a set of objects. When you analyze a change, DB2 Admin needs a base set of definitions for the change for the analyze process. DB2 Admin either extracts the object definitions from the catalog to use as the base version, which can be time consuming, or uses an existing version as the base version. You can specify that DB2 Admin uses an existing version when there are no prerequisite changes for the objects.

The CM - Manage Versions panel, which is shown in the following figure, is the main panel for managing versions:

```
 DB2 Admin ------------------ CM - Manage Versions --------------------- 16:59
 Option ===>


   1 - Display versions                                DB2 System: DB2X
   2 - Display version scopes                          DB2 SQL ID: ISTJE
   3 - Insert a version scope
   4 - Import version file







 Enter display selection criteria (Using a LIKE operator, criteria not saved):
 Name . . . . . . .                   > Created by . .          >
 Owner  . . . . . .          >          Altered by . .          >
 Created within . .                     Version ID . .
 Altered within . .
```

Figure 508. Manage Versions panel (ADB2C4)

Versions that have been generated in explicitly named data sets when you use DB2 Object Comparison Tool are not displayed because they are not stored in the Change Management database. When you use DB2 Object Comparison Tool and Change Management is enabled, you have the option of storing versions in data sets or in the Change Management database.

Displaying the versions

You can display the versions that are stored in the Change Management database.

About this task

To display the versions:

Procedure

1. Select option 4 on the Change Management (CM) panel to display the Manage Versions panel.
2. Optional: Enter the search criteria to filter or limit the versions that are displayed.
3. Select option 1 to display the Versions panel. The following figure shows an example of the Versions panel.

```
DB2 Admin ----------------- CM - Versions --------------- Row 1 to 8 of 64
Command ===>                                            Scroll ===> PAGE

Line commands:
 CH - Changes  PR - Promote  VS - Version scope  DEL - Delete  U - Update
 PT - Toggle protected status  I - Details on version

Sel          ID T Owner    Name               Comment
             * * *          *                  *
--- ----------- - -------- ----------------> -----------------------------
         290 B JOHNSON  HR_VER1
         291 D JOHNSON  ALT_ADD_COLUMN_AT_
         292 D KINCAID  CREATE_TB_TBTC
         394 B JOHNSON  HR_VER2
         295 D JOHNSON  ALT_MOD_TBTC
         305 B JOHNSON  HR_VER3
         334 D KINCAID  CREATE_TS_TSS1
         335 D KINCAID  CREATE_TS_TSS2
```

Figure 509. Versions panel (ADB2C41)

Results

You can issue a variety of line commands on the Versions panel for each version.
Commands are available to:
- See the changes that are associated with the version
- Promote the version
- See which scopes are associated with the version
- Set the protected status for the version
- Delete or update a version
- View details about the version

Versions that have been generated in explicitly named data sets when you use DB2
Object Comparison Tool are not displayed because they are not stored in the
Change Management database. When you use DB2 Object Comparison Tool and
Change Management is enabled, you have the option of storing versions in data
sets or in the Change Management database.

Creating a version from a version scope

You can create a version that is stored in the Change Management database from a
version scope.

About this task

To create a version from a version scope:

Procedure

1. Select option 4 on the Change Management (CM) panel to display the Manage
   Versions panel.
2. Select option 2 to display the Version Scopes panel, as shown in the following
   figure:

```
 DB2 Admin -------------- CM - Version Scopes --------------- Row 1 to 1 of 1
 Command ===>                                               Scroll ===> PAGE

 Line commands:
  VE - Versions  SO - Version scope objects  GV - Generate new version file
  INS - Insert  U - Update  DEL - Delete  I - Details on version scope
  CP - Copy privileges

 Sel         ID Owner    Name               Comment
             * *         *                  *
 --- ----------- -------- ------------------ ---------------------------------
             1 JOHNSON  HR_SCOPE           Scope for HR database
             2 JOHNSON  PAYROLL_SCOPE      Scope for payroll application
             8 KINCAID  MANU_SCOPE         Scope for manufacturing database
 ***************************** END OF DB2 DATA ******************************
```

Figure 510. Version Scopes panel (ADB2C42)

3. Specify the GV line command for the version scope for which you want to generate a version.
4. Specify an owner and name for the new version on the pop-up panel that is displayed. The JCL to create the version is displayed.
5. Review and submit the job to create the new version.

Creating a version when running a change

When you run a change, you can specify that a new base version is generated. The base version can be created before or after the change is implemented.

When you create a version, you must specify the method that is used to define the content of the base version:

AUTO
> Specify AUTO if you want the product to automatically determine the objects to put into the base version based on the objects that are being changed.

USER Specify USER if you want to provide a version scope that defines the object list. If you specify USER, ensure that an appropriate version scope for the version to be created exists.

You can use DB2 Admin online or CM batch mode to create a version when running a change.

Creating a version online
You can use the DB2 Admin online interface to create a version when running a change.

Procedure
1. Display the change to be run by selecting option 1 on the Change Management panel, and then select option 1 on the Manage Changes panel.
2. Issue the RN line command for the change that you want to run. When you issue the RN line command to run the change, specify the appropriate information on the Run a Change panel:
   - AUTO or USER in the **Generate base version before run** field to generate a new base version immediately before the change is implemented.
   - AUTO or USER in the **Generate base version after run** field to generate a new base version immediately after the change is implemented.

The CM - Specify Base Version Options panel (ADB2CEX3) is displayed after the Run a Change panel. In the following example, AUTO was chosen for the **Generate base version before run** option, and no base version was requested for the **Generate base version after run** option.

```
ADB2CEX3  ------------ CM - Specify Base Version Options ----------------
Command ===>

Commands: NEXT

Change . . . : DEMBIN2.V10DEVB CM PROC TEST

Specify the following for the base versions:

Existing base version action . .          (Auto,Replace; Default is Auto)

Base version before run:
  Scope Information: The object list will be automatically determined.
    Owner  . . . . . . . :          >           (? to lookup)
    Name . . . . . . . . :                          > (? to lookup)

  Version Information:
    Owner  . . . . . . . .          >           (? to lookup)
    Name . . . . . . . . .                          > (? to lookup)

Base version after run: A base version will not be generated after the run.
  Scope Information:
    Owner  . . . . . . . :          >           (? to lookup)
    Name . . . . . . . . :                          > (? to lookup)

  Version Information:
    Owner  . . . . . . . :          >           (? to lookup)
    Name . . . . . . . . :                          > (? to lookup)
```

Figure 511. CM - Specify Base Version Options

Attention: The base version will be overwritten if REPLACE is specified for the `Existing base version action` option. Specifying the base version owner and name is optional.

Creating a version using CM batch

You can use DB2 Admin change management batch mode to create a version when running a change.

Procedure

1. Modify the JCL template, setting parameters as appropriate for the type of version that you require.

   Specify the appropriate information in the **generate_base_version_before_run** and **generate_base_version_after_run** lines:

   - AUTO or USER in the **generate_base_version_before_run** line to generate a new base version immediately before the change is implemented.
   - AUTO or USER in the **generate_base_version_after_run** line to generate a new base version immediately after the change is implemented.

   The following JCL example imports a change, analyzes the change, and runs the change. A base version is created before and after the change is run. The base versions will be associated with the change.

   ```
   //BASEVF   JOB (DBA123,ICE,ICE,ICE),'SAMPLE',CLASS=B,
   //   MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=DBA123,TIME=(,30),
   //   REGION=0M
   //*
   /*JOBPARM S=SY4A
   //*
   ```

```
//LSCLIBS JCLLIB ORDER=ADB.DEVCUST.SAMP
//*
//GOCCM     EXEC GOCCM,SSID=DSNA,PLAN=ADB
//GOCCM.PARMS DD *
  change_name = 'S22957'
  ACTION_RUN_CHANGE = 'Y'
  generate_base_version_before_run = 'auto'
  generate_base_version_after_run = 'auto'
/*
//IMCHG001  DD *
--
  ALTER TABLE SCH123.EMP
    ADD COLUMN NEWCOL INT NOT NULL WITH DEFAULT;
/*
```

2. Run the JCL.

Generate DDL for the objects in a base version

You can generate DDL from a base version that is stored in Change Management.

Procedure

1. Display a list of base versions by using any of the following methods:

   - Enter the VE line command on a change to display a list of versions that are associated with the change (Admin option CM, 1, 1), and then issue the VE line command.

   - Enter the VE line command on a version scope to display a list of base versions that were created from the version scope (Admin option CM, 4, 2), and then issue VE line command.

   - Use Admin tool option CM, 4, 1 to display a list of versions.

2. Specify the DDL line command on the CM Versions panel to generate DDL for the objects in the base version, as shown in the following example:

```
ADB2C41 n ------------------ CM - Versions ------------------ Row 1 to 1 of 1
Command ===>                                            Scroll ===> CSR

Line commands:
 CH - Changes  PR - Promote  VS - Version scope  DEL - Delete  U - Update
 PT - Toggle protected status  I - Details on version  DDL - Generate DDL

Sel       ID T Owner   Name                          Comment
          * * *       *                              *
--- ----------- - -------- ---------------------------- --------------------
         3035 D DEMBIN2  SAMPLE
DDL      3037 B DEMBIN2  PRE-RUN 01
         3038 B DEMBIN2  POST-RUN 01
***************************** END OF DB2 DATA *****************************
```

This DDL line command is valid only for base versions (type=B) and not delta versions (type=D).

3. The CM Base Version DDL panel (ADB2C41E) is displayed with the base version owner and name fields filled in.

```
ADB2C41E  ---------------- CM - Base Version DDL ----------------------------
Command ===>

Specify the following options:
Base version:
  Owner  . . . . . . . . . DEMBIN2  >           (? to lookup)
  Name . . . . . . . . . . PRE-RUN 01                    >  (? to lookup)

SQL output data set:
  Prefix for data sets . . DEMBIN2
  Data set name  . . . . . BACKUP.DDL.PRERUN01
```

Regenerating Change Management versions containing LOBs

A new version of the records layout is created if LOB objects are involved in a
change management job.

About this task

This layout is not compatible with previous versions containing LOBs. Therefore,
you must regenerate older versions that contain LOB columns. You can identify
which change management base versions are affected by using this query:

```
SELECT OWNER,NAME,TYPE
   FROM ADB.ADBCVERSION V
     WHERE TYPE='B'
        AND EXISTS(
                   SELECT VERSIONID
                      FROM ADB.ADBCVERLINES VL
                      WHERE V.VERSIONID=VL.VERSIONID
                         AND VL.PREFIXGROUP=52)
```

You can identify the active CM changes that are affected by using this query:

```
SELECT C.OWNER,C.NAME,C.STATUS
   FROM ADB.ADBCVERSION V,ADB.ADBCHG C
     WHERE C.STATUS NOT IN ('COMPLETE','CANCELLED')
         AND V.TYPE='D'
         AND C.DELTAVERID =V.VERSIONID
         AND EXISTS(
                   SELECT VERSIONID
                      FROM ADB.ADBCVERLINES VL
                      WHERE V.VERSIONID=VL.VERSIONID
                         AND  C.DELTAVERID=VL.VERSIONID
                         AND VL.PREFIXGROUP=52)
```

To regenerate change management versions:

Procedure
1. Use the RST line command to restart INITIAL changes.
2. Make sure that RUNNING changes are completed.
3. Use the ST line command for all of the other changes that are listed and edit
   and SAVE one statement (without making any changes). The change is put into
   defined status and the change can be handled as usual.

Deleting a version

You cannot delete delta versions but you can delete base versions.

About this task

To delete a base version:

Procedure

1. Select option 4 on the Change Management (CM) panel to display the Manage Versions panel.
2. Select option 1 to display the Versions panel.
3. Issue the DEL line command for the version that you want to delete.
4. If you receive a message that indicates that the version is protected, issue the PT line command to remove the protected status and issue the DEL line command again. Delete the version only if you know that it is no longer needed.

Version scopes

A version scope defines the set of objects to include in the processing of a version.

A version scope determines the objects that are included in a version.

A version scope can be any set of objects, such as one or more databases, or a group of table spaces. Typically, you want to define scopes that identify all of the objects for an application or application area. For example, the scope for a human resources application should contain all the human resource databases.

After you create a version scope, you can create a base version for that set of objects.

A version scope must exist if you plan to create a new base version when you apply changes. If you have a new base version created when you run a change to reflect the object definitions after the changes, you must specify the version scope for the version.

Maintaining a version scope is a manual process, and you should ensure that the definition of the scope always includes all of the objects that you intend. For example, assume that you defined version scope SCOPE1 to include databases DB01 and DB02 and then created version BASE1. Later, you run CHANGE1, which creates a table in DB01 and creates a new database DB03, specifying to create a new base version BASE1 using SCOPE1. Database DB03 is not automatically added to SCOPE1.

The Manage Versions panel, which is shown in the following figure, is the main panel for working with version scopes:

```
DB2 Admin ------------------ CM - Manage Versions --------------------- 16:59
Option ===>


  1 - Display versions                              DB2 System: DB2X
  2 - Display version scopes                        DB2 SQL ID: ISTJE
  3 - Insert a version scope
  4 - Import version file






Enter display selection criteria (Using a LIKE operator, criteria not saved):
Name . . . . . . .                 > Created by . .         >
Owner  . . . . . .          >        Altered by . .         >
Created within . .                   Version ID . .
Altered within . .
```

Figure 512. Manage Versions panel (ADB2C4)

From the Manage Versions panel, you can display the existing version scopes to work with them or create a new version scope.

Creating a version scope

You can create a version scope.

About this task

To create a version scope:

Procedure

1. Select option 4 on the Change Management (CM) panel to display the Manage Versions panel.
2. Select option 3 on the Manage Versions panel to display the Insert Version Scope panel.
3. Specify a name and owner for the version scope, and, optionally, enter a comment for the version scope.
4. Press F3 to return to the Manage Versions panel.
5. Select option 2 to display the Version Scopes panel.
6. Specify the SO line command for the version scope that you created. The Version Scope Objects panel is displayed, as shown in the following figure:

```
DB2 Admin ----------- CM - Version Scope Objects ------------- Row 1 to 1 of 1
Command ===>                                             Scroll ===> PAGE

Version scope objects for scope "DBAUSER2"."NEWSCOPE"
Commands: SAVE
Line commands:
  I - Insert  D - Delete  R - Repeat

Sel T  Qual     Name               Oper.
    *  *        *                   *
--- -- -------> -----------------> ------

    ?  ?        ?
****************************** END OF DB2 DATA *******************************
```

Figure 513. Version Scope Objects panel (ADB2C4O)

7. Use the I line command to add each object that you want in the version scope, and specify the type of object, a qualifier, and a name for the object. You can also use the D line command to delete objects from the scope definition, and you can use the R line command to repeat a line to make it faster to define the objects in the scope.

   The values for the qualifier and name can contain zero or more of the following wildcard characters:
   • Minus sign (-) represents any single character.
   • Percent sign (%) or asterisk (*) represents one or more characters.
   • Any other character represents a single occurrence of itself.

   The rules for the wildcard characters follow the rules that are used for the LIKE predicate.

   See the online help for the Version Scope Objects panel for a description of the input fields, which includes a list of the types of objects that you can add.

   The following figure shows an example of a version scope definition.

```
DB2 Admin ----------- CM - Version Scope Objects ------------- Row 1 to 6 of 6
Command ===>                                             Scroll ===> PAGE

Version scope objects for scope "DBAUSER2"."NEWSCOPE"
Commands: SAVE
Line commands:
  I - Insert  D - Delete  R - Repeat

Sel T  Qual     Name               Oper.
    *  *        *                   *
--- -- -------> -----------------> ------

    DB          DBADB001
    TS DBADB002 TSAB%
```

Figure 514. Example of a version scope definition

8. Issue the SAVE primary command to save the definition of the scope.

Deleting a version scope

You can delete a version scope.

About this task

To delete a version scope:

Procedure

1. Select option 4 on the Change Management (CM) panel to display the Manage Versions panel.
2. Select option 2 to display the Version Scopes panel.
3. Issue the DEL line command for the version scope that you want to delete.

Displaying the version scopes

You can display the version scopes that are stored in the Change Management database.

About this task

To display the version scopes:

Procedure

1. Select option 4 on the Change Management (CM) panel to display the Manage Versions panel.
2. Select option 2 to display the Version Scopes panel. The following figure shows an example of the Version Scopes panel:

```
DB2 Admin --------------- CM - Version Scopes ---------------- Row 1 to 1 of 1
Command ===>                                            Scroll ===> PAGE

Line commands:
 VE - Versions  SO - Version scope objects  GV - Generate new version file
 INS - Insert  U - Update  DEL - Delete  I - Details on version scope

Sel        ID Owner    Name               Comment
           *  *        *                  *
--- ----------- -------- ------------------ ---------------------------------
            1 DBAUSER1 HR_SCOPE           Scope for HR database
            2 DBAUSER1 PAYROLL_SCOPE      Scope for payroll application
            8 DBAUSER3 MANU_SCOPE         Scope for manufacturing database
****************************** END OF DB2 DATA *******************************
```

Figure 515. Versions Scopes panel (ADB2C42)

Results

You can issue a variety of line commands on the Version Scopes panel for each version scope. Commands are available to:
- See which versions use the scope
- See which objects are in the scope
- Generate a new base version for the scope
- Insert, delete, or update a scope
- View details about who created the scope and when and who altered it last

Editing a version scope

You can add or delete objects from an existing scope.

About this task

To edit a version scope:

Procedure

1. Select option 4 on the Change Management (CM) panel to display the Manage Versions panel.

2. Select option 2 to display the Version Scopes panel.
3. Specify the SO line command for the version scope that you want to edit. The Version Scope Object panels, which shows the objects in the current definition, is displayed:

```
DB2 Admin ----------- CM - Version Scope Objects ------------- Row 1 to 6 of 6
Command ===>                                               Scroll ===> PAGE

Version scope objects for scope "DBAUSER1"."PAYROLL_SCOPE"
Commands: SAVE
Line commands:
  I - Insert  D - Delete  R - Repeat

Sel T  Qual      Name                Oper.
    *  *         *                   *
--- -- -------> -----------------> ------

    TB DBAUSER1 EMPLOYEE
    TB DBAUSER1 TIMECARDS
    FU VNDH01   FEDVALUES
```

Figure 516. Example of editing a version scope definition

4. Use the I and D line commands to insert or delete an object in the definition. Ensure that a type, a qualifier, and a name are specified for each object. You can also use the R line command to repeat a line to make it faster to define the objects in the scope.

   See the online help for the Version Scope Objects panel for a description of the input fields, which includes a list of the types of objects that you can add.
5. Issue the SAVE primary command to save the definition of the scope.

Importing a version file

You can import a version file to the change management database.

About this task

To import a version scope:

Procedure

1. Select option 4 on the Change Management (CM) panel to display the Manage Versions panel.
2. Select option 4 to display the Import Version File panel.
3. You can specify the following options on the panel:
   - **Version File DSN**: The data set name in which the version file to be imported is contained. The data set can be a stand-alone data set or a PDS with a member
   - **Owner**: The owner of the version to be added to the change management database
   - **Name**: The name of the version to be added to the change management database.
   - **Execution Mode**: Determines whether to import the version in the foreground (TSO) or in the background (batch).

```
ADB2C44 n --------------- Import Version File -------------- 08:05

Enter/verify the following:
  Version File DSN . . . . . . .
  Owner. . . . . . . . . . . . .         > (? to look up)
  Name . . . . . . . . . . . . .         > (? to look up)
  Execution Mode . . . . . . . .         Batch or TSO)
```

Figure 517. Example of importing a version file

Specifying a quick scope

A quick scope is similar in concept to a request parameter for the GEN operation. Whereas you use a request parameter to name the specific DB2 objects that a GEN operation generates, you can use a quick scope to identify the specific objects to compare in CM batch.

About this task

A quick scope has the same syntax and keywords as a request parameter. A quick scope supports the same types that are listed in Table 9 on page 226 in the "Generating SQL using wildcard characters" topic. In addition to those types, a quick scope supports the following type:

Table 24. The keyword values for quick scope

Object Type	TYPE	QUAL	NAME	Notes
DB2 Admin Version Scope	VSCOPE	owner	*name*	

Restriction: VSCOPE is only valid when used to specify a quick scope for the compare source or target in CM batch.

Tracking changes and changed objects

You can use the reporting feature in Change Management to display changes and changed objects and to check the history of changes.

You can use either the Changes panel or the Report Changes panel to display changes. The Report Changes panel, as shown in the following figure, is the main panel for displaying changed objects.

```
DB2 Admin ------------------ CM - Report Changes ---------------------- 20:41
Option ===>


   1 - Display changes                                  DB2 System: DB2X
   2 - Display changed objects                          DB2 SQL ID: ISTJE







Enter display selection criteria (Using a LIKE operator, criteria not saved):
Name . . . . . . .                    >    Created by . . . .        >
Owner  . . . . . .          >              Altered by . . . .        >
Type . . . . . . .                         Status . . . . . .
Created before . .                         Altered before . .
Created after  . .                         Altered after  . .
```

Figure 518. Report Changes panel (ADB2C6)

Displaying changes

You can display the changes that are stored in the Change Management database.

About this task

To display the Changes panel, which lists the changes:

Procedure

Select which method you want to use to display the Changes panel.
- Select option 1 on the Change Management (CM) panel to display the Manage Changes panel, and then select option 1.
- Select option 4 on the Change Management (CM) panel to display the Report Changes panel, and then select option 1.

When you use either action, you can specify search criteria to filter or limit the changes that are displayed, such as searching for changes by owner or status or searching for changes that were created or altered before or after a certain date. For example, to display all the changes that need to be analyzed, specify DEFINED in the **Status** field. See the online help for a description of the search fields.
The following figure shows an example of the Changes panel:

```
ADB2C11 n ----------------- CM - Changes -------------------- Row 1 to 7 of 7
 Command ===>                                               Scroll ===> CSR

 Commands: COMMENT  EXPORT
 Line commands:
  U - Update  AN - Analyze  RN - Run  VE - Versions  ST - Statements
  PQ - Prerequisites  IG - Ignores  MA - Masks  S - Show WSL  B - Checkpoint
  AT - List Associated Targets  ? - Show all line commands

 Sel      ID Owner    Name                        Type     Status   I Comment
          * *        *                            *        *        * *
 --- -------- -------- ------------------------- -------- -------- - ---------
         164 VIJAYAK  MTC1                        MULTI-TC INITIAL
         227 VIJAYAK  MTC1_CHG_MULTIPLE_DSNA      MULTI-TC DEFINED
  AT     287 J148286  MTC1                        MULTI-TC DEFINED
         423 VIJAYAK  MTC112                      MULTI-TC DEFINED
 ****************************** END OF DB2 DATA ******************************
```

Figure 519. Changes panel (ADB2C11)

What to do next

You can issue a variety of line commands on the Changes panel for each change.
Enter the ? line command to see a list of the available line commands. For
example, commands are available to:
- See the statements for a change
- See the prerequisites for a change.
- Analyze a change.
- Run a change.
- See the recover change for a change
- Recover a change

Displaying changed objects

You can display a list of objects that have changes.

About this task

The changes can be in any status and might not be complete.

To display changed objects:

Procedure

1. Select option 6 on the Change Management (CM) panel to display the Report
   Changes panel.
2. Optional: Use the fields at the bottom of the panel to specify the search criteria
   to filter or limit the objects that are displayed. For example, you can specify TB
   in the **Type** field to display only the tables that have changes. See the online
   help for a description of the search fields.
3. Select option 2 to display the Changed Objects panel. The following figure
   shows an example of the Changed Objects panel:

```
DB2 Admin -------------- CM - Changed Objects -------------- Row 1 to 13 of 14
Command ===>                                              Scroll ===> PAGE

Line commands:
 ST - Statements  CH - Change  CHA - All Changes  CHC - Completed Changes
 CHN - Not Completed Changes


        Change Change  Change              Object   Object
Sel     Sequence Owner  Name            O  Qualifier Name
               * *      *               *  *         *
--- ----------- -------- ---------------- -- --------- ------------------
            1 JOHNSON  EMP_CH1          TB DSNDV1DB  EMP
            1 JOHNSON  EMP_CH2          TB DSNDV1DB  EMP
            1 JOHNSON  DEPT_CH1         IX DSNDV1DB  DEPTNOIX
            1 JOHNSON  DEPT_CH2         TB DSNDV1DB  DEPT
            1 VNDH01   ACT_CH1          TS DSNDB04   ACT
            1 VNDH01   CRE_PTDB01       DB           PTDB01
            1 VNDH01   CRE_PTTS01       TS PTDB01    PTTS01
            1 VNDH01   CRE_EMPTB        TB TONELLO   PTTB01
            1 VNDH01   REC_CRE_PTDB01   DB           PTDB01
            1 VNDH01   REC_CRE_PTTS01   TS PTDB01    PTTS01
            1 VIJAYAK  EMP_C1           TB DSNDV2DB  EMP
            1 VNDH01   ACT_CH2          TB DSNDV1DB  ACT
            1 JOHNSON  ACT_CH3          TB DSNDV1DB  ACT
```

Figure 520. Changed Objects panel (ADB2C62)

4. Optional: Use the line commands to perform various actions on a changed
   object. For example, you can display all the completed changes for a particular
   object or you can get details on a particular change.

Chapter 22. Using masks

A *mask* (also called a *translation mask*) provides the ability to cause context-sensitive global changes to naming conventions and objects in generated SQL.

You can also use masks to overwrite the value of certain table space attributes.

For example, in DB2 Admin, you can specify masks to change names, objects, and qualifiers when you:
* Generate SQL to reverse engineer DB2 objects
* Clone a work statement list (WSL)
* Migrate DB2 object definitions, the data in those objects, or catalog statistics to other DB2 systems
* Import changes through Change Management.

You can define a mask either in a data set, or if Change Management is enabled on your system, in a table in the Change Management database. Masks that are specified on panels for reverse engineering SQL from the DB2 catalog, cloning WSLs, or migrating objects can be defined in a data set or in a table in the Change Management database. Masks that are specified when you import changes through Change Management must be defined in the Change Management database.

Tip: Consider managing all your masks through Change Management. The masks are easy to manage and recover because they are stored in a table in the Change Management database.

Topics:
* "Specifying a mask"
* "Mask definitions" on page 774

Specifying a mask

You can specify a mask when you generate SQL to reverse engineer DB2 objects, clone a work statement list (WSL), or migrate objects, data, or catalog statistics.

About this task

To specify a mask when you generate SQL to reverse engineer DB2 objects, clone a work statement list (WSL), or migrate objects, data, or catalog statistics:

Procedure
1. Specify `Yes` in the **Use Masking** field on the appropriate panel to display the Specify Mask panel. The following panels have the **Use Masking** field:
   * Generate SQL from DB2 catalog panel (ADB2GEN)
   * Clone Work Statement List panel (ADB2W1Q)
   * Migrate Parameters panel (ADB28M)

   The following figure shows the Specify Mask panel:

```
DB2 Admin --------------------- Specify Mask --------------------------
Option ===>


Mask Table Entry:
  Owner . .          >             (? to look up)
  Name  . .                    > (? to look up)
Data Set:
  Mask DSN  . .
Options:
  Edit Mask . .      (Yes/No)
```

Figure 521. Specify Mask panel (ADB2GENM)

> The **Mask Table Entry** fields that allow you to specify an owner and name are displayed only if Change Management is enabled on your system.

2. On the Specify Mask panel, specify the mask to use. Complete one of the following steps: To specify a mask that is defined in a data set:

   a. Specify the name of the data set that contains the masks to use. The mask data set must contain masks, must adhere to TSO naming conventions, and be one of the following types:

      • A fixed-block sequential data set
      • A member of a partitioned data set with a record length of 80 (RECFM=Fx,LRECL=80)

      If the specified data set name exists, it is reused. Otherwise, it is created.

   b. Specify Yes in the **Edit Mask** field if you want to edit the mask data set by using ISPF edit.

   To specify a mask that is defined in a table in the Change Management database:

   a. Specify the owner and the name of the mask in **Owner** and **Name** fields.

   b. Specify Yes in the **Edit Mask** field if you want to change the definition of the mask. When you specify Yes, the Mask Lines panel (ADB2C2L) is displayed.

      If you prefer to use ISPF edit to specify your edit masks, you can navigate to the Masks panel (ADB2C31), which lists the masks, and issue the E line command to display the mask definition in the Edit Masks panel.

      If you specify a mask that does not exist and you specify Yes in the **Edit Mask** field, the mask will be created for you in the Change Management database.

      If you specify both the owner and name of a mask table entry and a data set name, a data set is used.

Mask definitions

The mask definition describes how objects and names for objects are to be translated.

The mask definition also lets you overwrite the values of certain table space and index space attributes, including COMPRESS, DEFINE, DEFER, DSSIZE, PRIQTY, SECQTY, and SEGSIZE.

When you specify masks, they are processed in the order that you list them.

Note: The information in this topic about mask names, the mask hierarchy, how masks are applied, and performance is also applicable for masks that are defined in the Change Management database.

Mask definition syntax

You can specify one or more masks. Mask can contain generic specifications, which are expressed by using an asterisk.

If you are using a mask data set, to view or edit mask definitions, specify Yes in the **Edit Mask** field of the Specify Mask panel. When you press Enter, the mask definitions are displayed in ISPF Edit. The following figure shows mask definitions in the Edit Masks panel:

```
     ****** ***************************** Top of Data ******************************
==MSG>
==MSG>  Mask Syntax:
==MSG>    field:[qual<.name>:]inmask,outmask
==MSG>  Fields (hierarchy):
==MSG>    SINGLECH
==MSG>    COLNAME
==MSG>    NAME
==MSG>      DBNAME,TSNAME,IXNAME,UDFNAME,CONSNAME,
==MSG>      UDTNAME,COLLNAME,PKGNAME,PGMNAME,PLNNAME
==MSG>      DBRMNAME,STPNAME,SFNAME,TGNAME,GRPNAME,
==MSG>      VCATNAME,GBPNAME,TCNAME,PMNAME,MKNAME
==MSG>      SEQNAME,GVNAME
==MSG>      TBNAME
==MSG>        SYNNAME,ALNAME,VWNAME
==MSG>      BPNAME
==MSG>        TSBPNAME,IXBPNAME
==MSG>      SGNAME
==MSG>        TSSGNAME,IXSGNAME
==MSG>    AUTHID
==MSG>      SQLID
==MSG>      SCHEMA
==MSG>        IXSCHEMA,PMSCHEMA,MKSCHEMA,SETPATHSC
==MSG>        TGSCHEMA,UDTSCHEMA,SEQSCHEMA,STPSCHEMA
==MSG>        UDFSCHEMA,GVSCHEMA
==MSG>        TBSCHEMA
==MSG>          ALSCHEMA,VWSCHEMA
==MSG>      OWNER
==MSG>        DBOWNER,TSOWNER,IXOWNER,SGOWNER
==MSG>        PKGOWNER
==MSG>        TBOWNER
==MSG>          SYNOWNER
==MSG>      GRANTID
==MSG>        GRANTOR,GRANTEE
==MSG>      ROLE
==MSG>        DBROLE,TSROLE,TBROLE,IXROLE
==MSG>    XMLSCHID
==MSG>    WLMENV
==MSG>    LOCATION
==MSG>
==MSG>  Overwrite Syntax:
==MSG>    Field:inmask,Overwrite_value
==MSG>  Fields:              Overwrite values:
==MSG>    COMPRESS           YES,NO,REXX exit
==MSG>    SEGSIZE            n (4-64 must be multiple of 4),REXX exit
==MSG>    DSSIZE             nG,REXX exit
==MSG>    PRIQTY             n,n%,REXX exit (table spaces and indexes)
==MSG>      TSPRIQTY         n,n%,REXX exit (table spaces only)
==MSG>      IXPRIQTY         n,n%,REXX exit (indexes only)
==MSG>    SECQTY             n,n%,REXX exit (table spaces and indexes)
==MSG>      TSSECQTY         n,n%,REXX exit (table spaces only)
==MSG>      IXSECQTY         n,n%,REXX exit (indexes only)
==MSG>    DEFER              YES,NO,REXX exit (indexes only)
==MSG>    DEFINE             YES,NO,REXX exit (table spaces and indexes)
==MSG>      TSDEFINE         YES,NO,REXX exit (table spaces only)
==MSG>      IXDEFINE         YES,NO,REXX exit (indexes only)
==MSG>    HASHSPC            nK,nM,nG,REXX exit
==MSG>    TBINLOBL           n,REXX exit (tables only)
==MSG>    DTINLOBL           n,REXX exit (distinct types only)
==MSG>    AUDIT              CHANGES,ALL,NONE,REXX exit (tables only)
==MSG>    CLOSE              YES,NO,REXX exit (table spaces and indexes)
==MSG>      TSCLOSE          YES,NO,REXX exit (table spaces only)
==MSG>      IXCLOSE          YES,NO,REXX exit (indexes only)
==MSG>    TRACKMOD           YES,NO,REXX exit (table spaces only)
==MSG>    DCAPTURE           NONE,CHANGES,REXX exit (tables only)
==MSG>    FREEPG             n,REXX exit (table spaces and indexes)
==MSG>      TSFREEPG         n,REXX exit (table spaces only)
==MSG>      IXFREEPG         n,REXX exit (indexes only)
==MSG>    PCTFREE            n,REXX exit (table spaces and indexes)
==MSG>      TSPCTFREE        n,REXX exit (table spaces only)
==MSG>      IXPCTFREE        n,REXX exit (indexes only)
==MSG>    LOCKMAX            n,SYSTEM,REXX exit (table spaces only)
==MSG>    ERASE              YES,NO,REXX exit (table spaces and indexes)
==MSG>      TSERASE          YES,NO,REXX exit (table spaces only)
==MSG>      IXERASE          YES,NO,REXX exit (indexes only)
==MSG>    RESQNDROP          YES,NO,REXX exit (tables only)
```

Figure 522. Edit Masks panel, part 1

```
==MSG>    Notes:
==MSG>      - n is a integer value
==MSG>      - n% is the integer percentage of the current attribute value
==MSG>      - REXX exit takes format of REXX(myexit,val1,val2...valn) where
==MSG>        valn is the name of DB2 catalog field (such as PARTITIONS) or
==MSG>        a variable with numeric/string value (such as BPOOL= 'BP1').
==MSG>        + in col 72 indicates continuation of Rexx exit on next line
==MSG>      - To support/migrate DB2V8 masking input,OWNER,TBOWNER and
==MSG>        IXOWNER will mask both owner and schema fields.SCHEMA,
==MSG>        TBSCHEMA and IXSCHEMA will be applied to schema fields only.
==MSG>      - SINGLECH format is SINGLECH:<character>[,<escape character>]
==MSG>        where the single character in a mask specification represents
==MSG>        any character at that position. If the specified escape
==MSG>        character precedes the specified single character, then the
==MSG>        single character is treated as literal.
==MSG>      - The view, alias and synonym masks (both name and
==MSG>        schema/owner) will only apply to the CREATE statement for
==MSG>        these objects (e.g. VWNAME only valid for CREATE VIEW).
==MSG>        All other usages of these names and schemas are vague and
==MSG>        can refer also to table names and schemas.  These other
==MSG>        usages can only be masked by TBNAME for name and TBSCHEMA
==MSG>        for schema; therefore, it is recommended to use both VWNAME
==MSG>        and TBNAME if view names are being changed for both CREATE
==MSG>        VIEW statement and SQL that uses this view.
==MSG>      - Use caution when specifying mask field SEGSIZE. This mask
==MSG>        field might cause changes to the table space type. For
==MSG>        example, specifying the SEGSIZE mask might convert a
==MSG>        partitioned table space to a range-partitioned universal
==MSG>        table space (UTS). If a table in a UTS has a partitioned
==MSG>        index and the partitioned index needs to be recreated, DB2
==MSG>        might generate SQLCODE=-662 during execution.
==MSG>      - The following masks can not have the object-specific
==MSG>        qualifiers listed in the mask syntax:
==MSG>          NAME, SCHEMA, SETPATHSC, DBNAME, COLLNAME, SFNAME, GRANTID,
==MSG>          GRANTOR, GRANTEE, ROLE, DBROLE, TSROLE, TBROLE, IXROLE,
==MSG>          GBPNAME, TCNAME, XMLSCHID, AUTHID, SQLID, SGNAME, OWNER,
==MSG>          OWNER, BPNAME, PLNNAME and SINGLECH.
==MSG>    Mask examples:
==MSG>      OWNER:ABC*,DEF*
==MSG>      NAME:PRE*,NPRE*
==MSG>      XMLSCHID:PO1,PO2
==MSG>      WLMENV:WLM33,WLM44
==MSG>      LOCATION:LOC3*,LOCT*
==MSG>      SETPATHSC:SYSIBM,SYSFUN
==MSG>      SINGLECH:_
==MSG>      SINGLECH:_,+
==MSG>
==MSG>    Object-specific mask examples:
==MSG>      TBSCHEMA:CREATOR1.TB2:CREATOR1,NEW_CRE1
==MSG>      IXNAME:IXOWN*.IX3*:IX3*,IX4*
==MSG>      IXBPNAME:IXOWN1.INDX2:BP1,BP3
==MSG>
==MSG>    Overwrite examples:
==MSG>      COMPRESS:MYDB*.MYTS*,YES
==MSG>      SEGSIZE:MYDB*.MYTS*,8
==MSG>      DSSIZE:MYDB*.MYTS*,4G
==MSG>      PRIQTY:*.*,REXX(MYPRIQTY,DBNAME='MYDBTEST')
==MSG>      TSPRIQTY:MYDB*.MYTS*,30
==MSG>      IXPRIQTY:MYCR*.MYIX*,25%
==MSG>      IXSECQTY:MYCR*.MYIX*,REXX(MYSECQTY,IXNAME,IXCREATOR,PCT=20%)
==MSG>      DEFER:USER001.*IXNAME,NO
==MSG>      DEFINE:DBNAME*.*TSPC,REXX(MYDEFINE,DEFINE='YES')
==MSG>      HASHSPC:TBCREATOR.MYTBNAME,100M
==MSG>      TBINLOBL:TBCREATOR.MYTBNAME.COLNAME,16000
==MSG>      DTINLOBL:DTCRE*.DTNAME*,16000
==MSG>      IXCLOSE:MYCR*.MYIX*,NO
==MSG>      AUDIT:MYDB*.MYTB*,CHANGES
==MSG>      TRACKMOD:MYDB*.MYTS*,NO
==MSG>      DCAPTURE:TBCRE*.MYTB*,NONE
==MSG>      FREEPG:ABC*.DEF*,6
==MSG>      IXPCTFREE:IXSCH1.IXNAME1,9
==MSG>      LOCKMAX:DBTEST2.TSTEST2,SYSTEM
==MSG>      TSERASE:DBTEST1.TSTEST1,NO
==MSG>      RESONDROP:TBCRE*.MYTB*,NO
==MSG>
```

Figure 523. Edit Masks panel, Part 2

The message lines on the panel and Table 25 on page 780 list the available mask names for changing naming conventions and for overwriting table space and index space attribute values.

You can specify one or more masks. Masks can contain generic specifications, which are expressed by using an asterisk.

When you specify masks, they are processed in the order that you list them.

The syntax for specifying a mask to change naming conventions is shown in the following figure:

```
maskname: inputmask,outputmask
```

Figure 524. Translation mask syntax

Restriction: The maximum length allowed for input masks and output masks is 256 bytes each.

You use a plus sign (+) in column 72 to indicate continuation onto the next line.

The syntax for overwriting the value of a table space or index space attribute is shown in the following figure:

```
maskname: inputmask, overwrite_value
```

Figure 525. Overwrite syntax

The inputmask identifies the table space name or index space, and overwrite_value identifies the new value to use for the attribute. The value that you can specify for overwrite_value depends on the attribute, as shown in Figure 2. The value can be a direct value such as the YES or NO, an integer value (n), or an integer percentage of the current value (n%). The value can also be a REXX user exit that calculates a value. The maximum length allowed for input masks and a direct overwrite value is 256 bytes each. The maximum length allowed for specifying a REXX user exit and its input variables is 256 bytes. You use a plus sign (+) to indicate the continuation of a REXX user exit onto the next line. For more information about using a REXX user exit, see "Specifying a REXX user exit for the overwrite value" on page 787.

Notice that message lines on the panel show that the mask names have a hierarchy. For example, to change all DBNAMEs in the form of X* to Y*, specify the following:

```
DBNAME: X*,Y*

 or

NAME: X*,Y*
```

However, NAME is a grandparent in the hierarchy and, therefore, more general than DBNAME, which is a child. Therefore, using the higher-level mask changes all NAME masks.

Example 1: BPNAME has three levels: TSBPNAME, BPNAME and NAME. So, to translate a table space buffer pool name (TSBPNAME), you can use either TSBPNAME, BPNAME, or NAME. However, if you use BPNAME, all names that match the mask (table space and index space buffer pool) are translated.

Example 2: COLNAME has no levels and does not participate in a hierarchy. To translate a column name, you must use COLNAME.

Example 3: TSPRIQTY is second in the hierarchy of PRIQTY and TSPRIQTY. TSPRIQTY overwrites the PRIQTY for table spaces only; whereas PRIQTY overwrites the PRIQTY for both table spaces and index spaces.

Translation mask names

Table 25. Translation mask names

Name	Parent	Grandparent	Description
SINGLECH			Single character mask specification
COLNAME			Column name
		NAME	All names listed below
COLLNAME		NAME	Collection name
CONSNAME		NAME	Constraint name
DBNAME		NAME	Database name
DBRMNAME		NAME	DBRM name
GBPNAME		NAME	Group buffer pool name
GRPNAME		NAME	Group name
GVNAME		NAME	Name of global variable
IXNAME		NAME	Index name
PGMNAME		NAME	Program name; synonym for DBRM name
PKGNAME		NAME	Package name
PLNNAME		NAME	Plan name
SFNAME		NAME	Specific function name
STPNAME		NAME	Stored procedure name
TBNAME		NAME	Table, alias, synonym, and view names
TGNAME		NAME	Trigger name
TSNAME		NAME	Table space name
UDFNAME		NAME	User-defined function name
UDTNAME		NAME	User-defined data type name
VCATNAME		NAME	VCAT name
	SEQNAME	NAME	Sequence name mask
ALNAME	TBNAME	NAME	Name mask for aliases **Note:** This mask is valid only for CREATE statements where it is clear that the object is an alias.
SYNNAME	TBNAME	NAME	Name mask for synonyms

Table 25. Translation mask names (continued)

Name	Parent	Grandparent	Description
VWNAME	TBNAME	NAME	Name mask for views **Note:** This mask is valid only for CREATE statements where it is clear that the object is a view.
	SGNAME	NAME	All storage group names
IXSGNAME	SGNAME	NAME	Storage group name for indexes
TSSGNAME	SGNAME	NAME	Storage group name for table spaces
	BPNAME	NAME	All buffer pool names
IXBPNAME	BPNAME	NAME	Buffer pool name for indexes
TSBPNAME	BPNAME	NAME	Buffer pool name for table spaces
PMNAME		NAME	Masks the name of the row permission
MKNAME		NAME	Masks the name of the column mask
GRANTEE	GRANTID	AUTHID	Grantee
GRANTOR	GRANTID	AUTHID	Grantor
OWNER		AUTHID	Owner, creator, and so on. Masks the OWNER field.
DBOWNER	OWNER	AUTHID	Owner of the database
IXOWNER	OWNER	AUTHID	Owner of the index. Masks the index creator field (which is the OWNER of the index in DB2 V8, but is the SCHEMA of the index in DB2 V9)
TBOWNER	OWNER	AUTHID	Masks the table creator field (which is the OWNER of the table in DB2 V8, but is the SCHEMA of the table in DB2 V9)
SYNOWNER (subset of TBOWNER)	OWNER	AUTHID	Owner mask for synonyms
TSOWNER	OWNER	AUTHID	Owner of the table space
SCHEMA		AUTHID	Schema. Used to mask the SCHEMA field.
TBSCHEMA	SCHEMA	AUTHID	Masks the table creator field (which is the OWNER of table in DB2 V8, but the SCHEMA of table in DB2 V9)
ALSCHEMA (subset of TBSCHEMA)	SCHEMA	AUTHID	Schema mask for aliases **Note:** This mask is valid only for CREATE statements where it is clear that the object is an alias.
VWSCHEMA (subset of TBSCHEMA)	SCHEMA	AUTHID	Schema mask for views **Note:** This mask is valid only for CREATE statements where it is clear that the object is a view.
IXSCHEMA	SCHEMA	AUTHID	Masks the index creator field (which is the OWNER of index in DB2 V8, but the SCHEMA of index in DB2 V9)
GVSCHEMA	SCHEMA	AUTHID	Schema of global variable
SEQSCHEMA	SCHEMA	AUTHID	Sequence schema mask

Table 25. Translation mask names (continued)

Name	Parent	Grandparent	Description
SETPATHSC	SCHEMA	AUTHID	Schema name mask for SET CURRENT PATH schema statement
STPSCHEMA	SCHEMA	AUTHID	Stored procedure schema mask
TGSCHEMA	SCHEMA	AUTHID	Trigger schema mask
UDFSCHEMA	SCHEMA	AUTHID	Function schema mask
UDTSCHEMA	SCHEMA	AUTHID	
XMLSCHID			Masks the registered XML schema name in an XML-type modifier
WLMENV			WLM (Workload Manager) environment name mask
LOCATION			LOCATION mask, where "LOCATION" is the first of a three-part name, as in: *LOCATION*.schema.name
PMSCHEMA	SCHEMA	AUTHID	Masks the schema of the row
MKSCHEMA	SCHEMA	AUTHID	Masks the schema of the column mask
SQLID		AUTHID	Current SQLID
COMPRESS			Whether a table space or table space partition is compressed
SEGSIZE			Number of pages in each segment of a segmented table space
DSSIZE			Maximum size in gigabytes for each partition in a partitioned table space
	PRIQTY		Minimum primary space allocation for a DB2-managed data set for table spaces and index spaces
IXPRIQTY	PRIQTY		Minimum primary space allocation for a DB2-managed data set for index spaces
TSPRIQTY	PRIQTY		Minimum primary space allocation for a DB2-managed data set for table spaces
	SECQTY		Minimum secondary space allocation for a DB2-managed data set for table spaces and index spaces
IXSECQTY	SECQTY		Minimum secondary space allocation for a DB2-managed data set for index spaces
TSSECQTY	SECQTY		Minimum secondary space allocation for a DB2-managed data set for table spaces
DEFER			Whether to build the index during when the CREATE INDEX statement is run
	DEFINE		Whether the underlying data sets for the table space or index space are created when the object is created or when data is inserted into the object
IXDEFINE	DEFINE		Whether the underlying data sets for the index space is created when the index space is created or when data is inserted into the index space

Table 25. Translation mask names (continued)

Name	Parent	Grandparent	Description
TSDEFINE	DEFINE		Whether the underlying data sets for the table space is created when the table space is created or when data is inserted into the table space
TCNAME		NAME	Masks a trusted context name
ROLE	AUTHID		Mask a role name
DBROLE	ROLE	AUTHID	Masks a role associated with a database
TSROLE	ROLE	AUTHID	Masks a role associated with a table space
TBROLE	ROLE	AUTHID	Masks a role associated with a table
IXROLE	ROLE	AUTHID	Masks a role associated with an index
HASHSPC			To overwrite HASH SPACE integer
TBINLOBL			To overwrite INLINE LENGTH integer value for tables
DTINLOBL			To overwrite INLINE LENGTH integer value for distinct types
AUDIT			Records the value of the AUDITING option for a table
TRACKMOD			Whether to track the page modifications in the space map
DCAPTURE			Records the value of DATA CAPTURE option for a table
CLOSE			Specifies whether the data set is eligible to be closed
TSCLOSE			Specifies whether the data set is eligible to be closed
IXCLOSE			Specifies whether the data set is eligible to be closed
FREEPG			Number of pages loaded before a page is left as free space
TSFREEPG	FREEPG		Number of pages loaded before a page is left as free space for tablespaces
IXFREEPG	FREEPG		Number of pages loaded before a page is left as free space for indexes
PCTFREE			Percentage of each page left as free space
TSPCTFREE	PCTFREE		Percentage of each page left as free space for tablespaces
IXPCTFREE	PCTFREE		Percentage of each page left as free space for indexes
LOCKMAX			The maximum number of locks per user to acquire for the table or table space before escalating to the next locking level
ERASE			Indicates whether the DB2-managed data sets are to be erased
TSERASE	ERASE		Indicates whether the DB2-managed data sets are to be erased for tablespaces

Table 25. Translation mask names (continued)

Name	Parent	Grandparent	Description
IXERASE	ERASE		Indicates whether the DB2-managed data sets are to be erased for indexes
RESONDROP			Overwrites RESTRICT ON DROP attribute for tables

Note: The DBROLE, TSROLE, TBROLE, and IXROLE masks are not currently used.

The following mask names are used only when work statement lists (WSLs) are cloned. If specified, they have no affect in GEN, migrate, or importing changes.

DBRMNAME
> DBRM name. Used for BIND commands.

GBPNAME
> Group buffer pool name.

SFNAME
> Specific function name.

SQLID
> Needed by cloning for masking already generated SET CURRENT SQLID statements.
>
> Even if GEN and migrate generate SET CURRENT SQLID = <sqlid> statements, the SQLID mask is not used to mask the <sqlid>. The <sqlid> in these statements originates from field values in the DB2 Catalog and these values are masked before the SET statement is generated.
>
> **Example:** CREATE SYNONYM requires a SET CURRENT SQLID statement to set the current sqlid to the synonym owner (creator). The OWNER mask is used to mask the synonym owner before the SET statement is generated.

The following mask names have no affect when WSLs are cloned:
- DBOWNER
- TSOWNER
- SGOWNER
- PKGOWNER

Specifying a mask that applies only to specific objects (object-specific)

The effects of some masks are too general for all situations. For example, the IXBPNAME mask changes the name of every instance of the bufferpool. If you need to change a bufferpool for only one index, you can use object-specific masking. Consider the following IXBPNAME mask:

```
IXBPNAME:IXOWN1.IX2:BP1,BP3
```

. With this mask, only the index IXOWN1.IX2 has its bufferpool changed to BP3.

The syntax for specifying an object specific mask is shown in the following figure:

```
maskname:qual.name:inputvalue,outputmask
```

Figure 526. Object-specific mask syntax

The *qual* element is optional and when provided is a qualifier for the name of the object. For example, TBNAME:CREATOR1.TB2:CREATOR1,NEW_CRE1 means the mask applies only to the CREATOR1.TB2 table. Table 26 lists all of the object-specific masks.

Remember:

- When you use object-specific masking, the input mask can be greater than 256 bytes.
- The *name* element does not always refer to the name of the masked item. For example, for the IXSGNAME mask, the *Name* refers to the index name not the storage group name.

Table 26. Object-specific masks and the objects they affect

Name	Syntax
ALNAME	ALNAME:*alias_schema.alias_name:current_alname,new_alname*
ALSCHEMA	ALSCHEMA:*alias_schema.alias_name:current_alschema,new_alschema*
COLNAME	COLNAME:*table_schema.table_name:current_colname,new_colname*
CONSNAME	CONSNAME:*table_schema.table_name:current_consname,new_consname*
DBOWNER	DBOWNER:*database_name:current_dbowner,new_dbowner*
DBRMNAME$_1$	DBRMNAME:*stp_schema.stp_name:current_dbrmname,new_dbrmname*
DBRMNAME	DBRMNAME:*udf_schema.udf_name:current_dbrmname,new_dbrmname*
DBRMNAME	DBRMNAME:*table_schema.table_name:current_dbrmname,new_dbrmname*
GRPNAME	GRPNAME:*database_name:current_grpname,new_grpname*
GVNAME	GVNAME:*gv_schema.gv_name:current_gvname,new_gvname*
GVSCHEMA	GVSCHEMA:*gv_schema.gv_name:current_gvschema,new_gvschema*
IXBPNAME$_2$	IXBPNAME:*index_schema.index_name:current_bpname,new_bpname*
IXBPNAME	IXBPNAME:*database_name:current_db_indexbpname,new_db_indexbpname*
IXNAME	IXNAME:*index_schema,index_name:current_ixname,new_ixname*
IXOWNER	IXOWNER:*index_schema.index_name:current_ixowner,new_ixowner*
IXSCHEMA	IXSCHEMA:*index_schema.index_name:current_ixschema,new_ixschema*
IXSGNAME	IXSGNAME:*index_schema.index_name:current_ixsgname,new_ixsgname*
LOCATION	LOCATION:*schema_name.obj_name:current_location,new_location*
MKNAME	MKNAME:*mask_schema.mask_name:current_maskname,new_maskname*
MKSCHEMA	MKSCHEMA:*mask_schema.mask_name:current_mkschema,new_mkschema*
PGMNAME$_1$	PGMNAME:*stp_schema.stp_name:current_pgmname,new_pgmname*
PGMNAME	PGMNAME:*udf_schema.udf_name:current_pgmname,new_pgmname*
PGMNAME	PGMNAME:*table_schema.table_name:current_pgmname,new_pgmname*
PKGNAME	PKGNAME:*collection_id.package_name:current_pkgname,new_pkgname*
PKGOWNER	PKGOWNER:*collection_id.package_name:current_packageowner,new_packageowner*
PMNAME	PMNAME:*pm_schema.pm_name:current_pmname,new_pmname*
PMSCHEMA	PMSCHEMA:*pm_schema.pm_name:current_pmschema,new_pmschema*
SEQNAME	SEQNAME:*seq_schema.seq_name:current_seqname,new_seqname*
SEQSCHEMA	SEQSCHEMA:*seq_schema.seq_name:current_seqschema,new_seqschema*
SGOWNER	SGOWNER:*stogroup_name:current_stogroupowne*r, *new_stogroupowner*
STPNAME	STPNAME:*stp_schema.stp_name:current_stpname,new_stpname*

Table 26. Object-specific masks and the objects they affect (continued)

Name	Syntax
STPSCHEMA	STPSCHEMA:*stp_schema.stp_name*:current_stpschema,new_stpschema
SYNNAME	SYNNAME:*synonym_owner.synonym_name*:current_syname,new_syname
SYNOWNER	SYNOWNER:*synonym_owner.synonym_name*:current_synowner,new_synowner
TBNAME	TBNAME:*table_schema.table_name*:current_tbname,new_tbname
TBOWNER	TBOWNER:*table_schema.table_name*.current_tbowner,new_tbowner
TBSCHEMA	TBSCHEMA:*table_schema.table_name*:current_tbschema,new_tbschema
TGNAME	TGNAME:*trigger_schema.trigger_name*:current_tgname,new_tgname
TGSCHEMA	TGSCHEMA:*trigger_schema.trigger_name*:current_tbschema,new_tgschema
TSBPNAME₂	TSBPNAME:*database_name.tablespace_name*:current_tspbname,new_tsbpname
TSBPNAME	TSBPNAME:*database_name*:current_dbbpname,new_dbbpname
TSNAME	TSNAME:*database_name.tablespace_name*:current_tsname,new_tsname
TSOWNER	TSOWNER:*database_name.tablespace_name*:current_tsowner,new_tsowner
TSSGNAME₂	TSSGNAME:*database_name.tablespace_name*:current_tssgname,new_tssgname
TSSGNAME	TSSGNAME:*database_name*:current_dbsgname,new_dbsgname
UDFNAME	UDFNAME:*udf_schema.udf_name*:current_udfname,new_udfname
UDFSCHEMA	UDFSCHEMA:*udf_schema.udf_name*:current_udfschema,new_udfschema
UDTNAME	UDTNAME:*udt_schema.udt_name*:current_udtname,new_udtname
UDTSCHEMA	UDTSCHEMA:*udt_schema.udt_name*:current_udtschema,new_udtschema
VCATNAME	VCATNAME:*stogroup_name*:current_vcatname, new_vcatname
VCATNAME₁	VCATNAME:*schema.obj_name*:current_vcatname,new_vcatname
VWNAME	VWNAME:*view_schema.view_name*:current_vwname,new_vwname
VWSCHEMA	VWSCHEMA:*view_schema.view_name*:current_vwschema,new_vwschema
WLMENV₁	WLMENV:*udf_schema.udf_name*:current_wlmenvname.new_wlmenvname
WLMENV	WLMENV:*stp_schema.stp_name*:current_wlmenvname,new_wlmenvname

Note:

1. The DBRMNAME, PGMNAME, and VCATNAME masks can be used for more than one object type.
2. The IXBPNAME, TSPBNAME, and TSSGNAME masks can be used for both object-level and database-level versions of the names.

Restriction:

The following masks cannot have object-specific qualifiers:
- NAME
- SCHEMA
- SETPATHSC
- DBNAME
- COLLNAME
- SFNAME
- GRANTID
- GRANTOR
- GRANTEE
- ROLE
- DBROLE

- TSROLE
- TBROLE
- IXROLE
- GBPNAME
- TCNAME
- XMLSCHID
- AUTHID
- SQLID
- SGNAME
- OWNER
- BPNAME
- PLNNAME
- SINGLECH

Specifying a REXX user exit for the overwrite value

You can use a REXX user exit to specify the overwrite value for the table space and index space attributes for COMPRESS, SEGSIZE, DEFER, DEFINE, DSSIZE, PRIQTY, TSPRIQTY, IXPRIQTY, SECQTY, TSSECQTY, IXSECQTY, FREEPG, TSFREEPG, IXFREEPG, PCTFREE, TSPCTFREE, IXPCTFREE, LOCKMAX, ERASE, TSERASE, IXERASE, TRACKMOD, DCAPTURE, AUDIT, CLOSE, TSCLOSE, IXLCOSE, and RESONDROP.

About this task

Using a REXX user exit to calculate the value enables you to define and write your own overwrite rules to provide for additional flexibility and customization.

To specify a REXX user exit as the overwrite value in your mask definition:

Procedure

1. Ensure that DB2 Admin was customized to define the data set names for the REXX user exit libraries. You run Tools Customizer to customize DB2 Admin.
2. Ensure that a REXX user exit to calculate and return a valid value for the overwrite value has been defined and stored in the appropriate REXX user exit library. An example of a REXX user exit is shipped in SAMP library ADBDSIZE. The name of the REXX user exit in this sample is defined as MYDSSIZE, and the user exit calculates and returns a value that is to be used as the overwrite value for DSSIZE.
3. Define the mask definition with the correct syntax for calling the REXX user exit, which includes specifying the name of the REXX user exit and the input variables to pass as arguments to the user exit:
   REXX(*execname*,*val1*,*val2*,*...valn*).Each input variable must be the name of a DB2 catalog column or a variable name with a numeric or string value, where the variable name is the name of a DB2 catalog column. The following list shows some examples of the syntax that can be used on the Edit Mask panel to define overwrite values that are calculated by a REXX user exit:

```
DSSIZE: MYDB*. MYTS*, DSSIZE(MYDSSIZE, PARTITIONS, BPOOL)
DSSIZE: MYDB*. MYTS*, DSSIZE(MYDSSIZE, PARTITIONS=30, BPOOL='BP1')
PRIQTY: MYDB*. MYTS*, REXX(MYPQTY, DBNAME, TSNAME, PCT= 20%)
DEFINE: MYDB*. MYTS*, REXX(MYDEFINE, DEFINE='YES')
DEFER: MYDB*. MYTS*, REXX(MYDEFER, DEFER='NO')
COMPRESS: MYDB*. MYTS*, REXX(MYCOMP,TSNAME,DBAME,COMPRESS)
SEGSIZE: MYDB*. MYTS*, REXX(MYSEG, NAME,DBNAME,SEGSIZE)
```

```
| FREEPG:*.*,REXX(MYFREEPG,DBNAME,TSNAME,IXCREATOR,IXNAME)
| PCTFREE:*.*,REXX(MYPCT,DBNAME='MYDBTEST',TSNAME='MYTSTEST',IXCREATOR='MYIXSCH1',IXNAME='MYIXNAM1')
| LOCKMAX:DBTEST3.TSTEST3,REXX(MYLOCKM,NAME,DBNAME)
| ERASE:*.*,REXX(MYERASE,NAME,DBNAME,ERASERULE)
| RESONDROP:TBCRE*.TB*,REXX(MYRODEX,DBNAME,TSNAME)
```

The input values are passed to the REXX user exit in an argument list where
the REXX user exit uses the arguments to perform the calculations and return
the value that is to be used as the overwrite value. If one of the input variables
is not provided in the proper context, a minus sign (-) is passed to the REXX
user exit as the argument.

Restriction: When you specify the input values for a REXX user exit in the
mask definition that is to be used for WSL cloning or the import function in
Change Management, specify the input variables as DB2 catalog names that are
set to numeric or string variables. If you specify a catalog name only, the
variable is passed as a minus sign (-), and the REXX user exit will return a
value of a minus sign (-), which indicates that masking was not applied.

If the REXX user exit does not return a valid value for the overwrite value,
masking is not applied, and DB2 Admin processes the next definition in the
mask file.

DB2 catalog records that have default masks

The table in this topic shows the catalog columns in DB2 catalog records that have
masks applied before the SQL is created.

Table 27. Mask application details

DB2 Catalog record	Catalog column	Most specific mask names	Comments
SYSAUXRELS	TBNAME	TBNAME	
	TBOWNER	OWNER	
		TBOWNER	
	COLNAME	COLNAME	
	AUXTBNAME	TBNAME	
	AUXTBOWNER	OWNER	
		TBOWNER	
SYSCHECKS	TBOWNER	OWNER	
		TBOWNER	
	CREATOR	OWNER	
	TBNAME	TBNAME	
	CHECKCONDITION	COLNAME	Mask column names

Table 27. Mask application details (continued)

DB2 Catalog record	Catalog column	Most specific mask names	Comments
SYSCOLAUTH	GRANTOR	GRANTOR	
	TNAME	TBNAME	
	CREATOR	OWNER	
		TBOWNER	
	GRANTEE	PKGNAME	If grantee is package
		PLNNAME	If grantee is plan
		GRANTEE	If grantee is an authorization ID
	COLNAME	COLNAME	
	COLLID	COLLNAME	If grantee is package
SYSCOLUMNS	NAME	COLNAME	
	TBNAME	TBNAME	
	TBCREATOR	OWNER	If schema not SYSIBM
		TBOWNER	If schema not SYSIBM
	TYPENAME	UDTNAME	If schema not SYSIBM
	TYPESCHEMA	SCHEMA	
	TBOWNER	OWNER	
		TBOWNER	
	CREATOR	OWNER	
	TBNAME	TBNAME	
	CHECKCONDITION	COLNAME	Mask column names
	LENGTH	TBINLOBL	If Length is greater than 4 for INLINE LOB columns
SYSCONTROLS	SCHEMA	PMSCHEMA	If control_type is row permission
	NAME	PMNAME	
	SCHEMA	MKSCHEMA	If control_type is column mask
	NAME	MKNAME	
SYSDATABASE	NAME	DBNAME	
	CREATOR	OWNER	
		DBOWNER	
	STGROUP	TSSGNAME	
	BPOOL	TSBPNAME	
	GROUP_MEMBER	GRPNAME	
	INDEXBP	IXBPNAME	

Table 27. Mask application details (continued)

DB2 Catalog record	Catalog column	Most specific mask names	Comments
SYSDATATYPES	SCHEMA	SCHEMA	
	OWNER	OWNER	
	NAME	UDTNAME	
	INLINE_LENGTH	DTINLOB	If distinct type is based on LOB source type
SYSDBAUTH	GRANTOR	GRANTOR	
	GRANTEE	GRANTEE	
	NAME	DBNAME	
SYSFIELDS	TBCREATOR	OWNER	
		TBOWNER	
	TBNAME	TBNAME	
	NAME	COLNAME	
	FLDPROC	PGMNAME	
SYSFOREIGNKEYS	CREATOR	OWNER	
		TBOWNER	
	TBNAME	TBNAME	
	RELNAME	NAME	
	COLNAME	COLNAME	
SYSVARIABLES	SCHEMA	GVSCHEMA	
	NAME	GVNAME	
	OWNER	OWNER	
SYSVARIABLEAUTH	GRANTOR	GRANTOR	
	GRANTEE	GRANTEE	
	SCHEMA	GVSCHEMA	
	NAME	GVNAME	
SYSINDEXES	NAME	IXNAME	
	CREATOR	OWNER	
		IXOWNER	
	TBNAME	TBNAME	
	TBCREATOR	OWNER	
		TBOWNER	
	DBNAME	DBNAME	
	BPOOL	IXBPNAME	
	CLOSERULE	CLOSE	
		IXCLOSE	
	ERASERULE	ERASE	
		IXERASE	

Table 27. Mask application details (continued)

DB2 Catalog record	Catalog column	Most specific mask names	Comments
SYSINDEXPART	FREEPAGE	FREEPG	
		IXFREEPG	
	IXNAME	IXNAME	
	IXCREATOR	OWNER	
		IXOWNER	
	PCTFREE	PCTFREE	
		IXPCTFREE	
	STORNAME	IXSGNAME	
	VCATNAME	VCATNAME	
	PQTY	PRIQTY	
		IXPRIQTY	
	SQTY	SECQTY	
		IXSECQTY	
SYSKEYCOLUSE	TBCREATOR	OWNER	
		TBOWNER	
	TBNAME	TBNAME	
	COLNAME	COLNAME	
SYSKEYS	IXNAME	IXNAME	
	IXCREATOR	OWNER	
		IXOWNER	
	COLNAME	COLNAME	
SYSPACKAGE	COLLID	SCHEMA	Trigger package
	NAME	TGNAME	Trigger package
	COLLID	COLLNAME	Normal package
	NAME	PKGNAME	Normal package
	OWNER	PKGOWNER	
	CREATOR	PKGOWNER	
	QUALIFIER	SCHEMA	
	PATHSCHEMAS	SCHEMA	Applied to each schema

Table 27. Mask application details (continued)

DB2 Catalog record	Catalog column	Most specific mask names	Comments
SYSPACKDEP	BNAME	TBNAME	btype 'T','V','A','S','G'
		IXNAME	btype 'I'
		TSNAME	btype 'R','P'
		UDFNAME	btype 'F'
		STPNAME	btype 'O'
		NAME	btype none of the above
	BCREATOR	DBNAME	btype 'R','P'
		SCHEMA	btype 'F','O'
		TBOWNER	btype 'T','V','A','S','G'
		IXOWNER	btype 'I'
		OWNER	btype none of the above
	DNAME	TGNAME	Trigger package
	DCOLLID	SCHEMA	Trigger package
	DNAME	PKGNAME	Normal package
	DCOLLID	COLLNAME	Normal package
	DOWNER	OWNER	
SYSPARMS	SCHEMA	SCHEMA	
	OWNER	OWNER	
	NAME	UDFNAME	UDF
	SPECIFICNAME	UDFNAME	UDF
	NAME	STPNAME	Stored procedure
	SPECIFICNAME	STPNAME	Stored procedure
	TYPESCHEMA	SCHEMA	If schema not SYSIBM
	TYPENAME	UDTNAME	If schema not SYSIBM
SYSPLAN	NAME	PLNNA	
	CREATOR	OWNER	
	QUALIFIER	OWNER	
	PATHSCHEMAS	SCHEMA	Applied to each schema
SYSPLANDEP	BNAME	TBNAME	btype 'T','V','A','S','G'
		IXNAME	btype 'I'
		TSNAME	btype 'R','P'
		UDFNAME	btype 'F'
		STPNAME	btype 'O'
		NAME	btype none of the above
	BCREATOR	DBNAME	btype 'R','P'
		SCHEMA	btype 'F','O'
		TBOWNER	btype 'T','V','A','S','G'
		IXOWNER	btype 'I'
	DNAME	PLNNAME	

Table 27. Mask application details (continued)

DB2 Catalog record	Catalog column	Most specific mask names	Comments
SYSRELS	CREATOR	OWNER	
		TBOWNER	
	TBNAME	TBNAME	
	RELNAME	NAME	
	REFTBNAME	TBNAME	
	REFTBCREATOR	OWNER	
		TBOWNER	
	IXOWNER	OWNER	If non-blank
		IXOWNER	If non-blank
	IXNAME	IXNAME	If non-blank
SYSRESAUTH	GRANTOR	GRANTOR	
	GRANTEE	GRANTEE	
	NAME	TSBPNAME	obtype 'B'
		COLLNAME	obtype 'C'
	QUALIFIER	SCHEMA	obtype 'D'
	NAME	UDTNAME	obtype 'D'
	QUALIFIER	DBNAME	obtype 'R'
	NAME	TSNAME	obtype 'R'
		TSSGNAME	obtype 'S'
		NAME	obtype 'J'
SYSROUTINEAUTH	GRANTOR	GRANTOR	
	GRANTEE	PKGNAME	If package
	COLLID	COLLNAME	If package
	GRANTEE	PLNNAME	If plan
		GRANTEE	If authorization ID GRANTEE
	SCHEMA	SCHEMA	
SYSROUTINES	SCHEMA	SCHEMA	
	OWNER	OWNER	
	NAME	UDFNAME	UDF
	SPECIFICNAME	UDFNAME	UDF
	NAME	STPNAME	stored procedure
	SPECIFICNAME	STPNAME	stored procedure
	COLLID	COLLNAME	
	SOURCESCHEMA	SCHEMA	
	EXTERNAL_NAME	PGMNAME	
	JARSCHEMA	SCHEMA	
	JAR_ID	NAME	

Table 27. Mask application details (continued)

DB2 Catalog record	Catalog column	Most specific mask names	Comments
SYSSCHEMAAUTH	GRANTOR	GRANTOR	
	GRANTEE	GRANTEE	
	SCHEMANAME	SCHEMA	
SYSSEQUENCEAUTH	GRANTOR	GRANTOR	
	GRANTEE	GRANTEE	
	SCHEMA	SCHEMA	
	NAME	NAME	
SYSSEQUENCES	SCHEMA	SCHEMA	
	OWNER	OWNER	
	NAME	NAME	
SYSSTOGROUP	NAME	TSSGNAME	
	CREATOR	CREATOR	
	VCATNAME	VCATNAME	
SYSSYNONYMS	NAME	TBNAME	
	CREATOR	OWNER	
	TBNAME	TBNAME	
	TBCREATOR	OWNER	
		TBOWNER	
SYSTABAUTH	GRANTOR	GRANTOR	
	GRANTEE	PKGNAME	If package
	COLLID	COLLNAME	If package
	GRANTEE	PLNNAME	If plan
		GRANTEE	If authid grantee
	BNAME	DBNAME	
	SCREATOR	OWNER	
	STNAME	TBNAME	
		TBOWNER	
	TCREATOR	OWNER	
		TBOWNER	
	TTNAME	TBNAME	
SYSTABCONST	TBCREATOR	OWNER	
		TBOWNER	
	TBNAME	TBNAME	
	CREATOR	OWNER	
	IXOWNER	OWNER	
		IXOWNER	
	IXNAME	IXNAME	

Table 27. Mask application details (continued)

DB2 Catalog record	Catalog column	Most specific mask names	Comments
SYSTABLEPART	TSNAME	TSNAME	
	DBNAME	DBNAME	
	IXNAME	IXNAME	
	IXCREATOR	OWNER	
		IXOWNER	
	STORNAME	TSSGNAME	
	VCATNAME	VCATNAME	
	COMPRESS	COMPRESS	
	HASHSPACE	HASHSPC	
	TRACKMOD	TRACKMOD	
	PQTY	PRIQTY	
		TSPRIQTY	
	SQTY	SECQTY	
		TSSECQTY	
	FREEPAGE	FREEPG	
		TSFREEPG	
	PCTFREE	PCTFREE	
		TSPCTFREE	
SYSTABLES	NAME	TBNAME	
	CREATOR	OWNER	
		TBOWNER	
	DBNAME	DBNAME	
	TSNAME	TSNAME	
	EDPROC	PGMNAME	
	VALPROC	PGMNAME	
	TBCREATOR	OWNER	
		TBOWNER	
	TBNAME	TBNAME	
	AUDITING	AUDIT	
	DATACAPTURE	DCAPTURE	
	CLUSTERTYPE	RESONDROP	

Table 27. Mask application details (continued)

DB2 Catalog record	Catalog column	Most specific mask names	Comments
SYSTABLESPACE	NAME	TSNAME	
	CREATOR	OWNER	
		TSOWNER	
	DBNAME	DBNAME	
	BPOOL	TSBPNAME	
	SEGSIZE	SEGSIZE	
	LOCKMAX	LOCKMAX	
	CLOSERULE	CLOSE	
		TSCLOSE	
	ERASERULE	ERASE	
		TSERASE	
SYSTRIGGERS	NAME	TGNAME	
	SCHEMA	SCHEMA	
	OWNER	OWNER	
	TBNAME	TBNAME	
	TBOWNER	OWNER	
		TBOWNER	
	TEXT	SCHEMA	Mask trigger name
		TGNAME	Mask tab/view/synonym
		OWNER	Mask UDT/UDF/STP
		TBNAME	
		SCHEMA	
		UDTNAME	
		UDFNAME	
		STPNAME	
		COLNAME	Mask column name

Table 27. Mask application details (continued)

DB2 Catalog record	Catalog column	Most specific mask names	Comments
SYSVIEWS	NAME	TBNAME	
	CREATOR	OWNER	
		TBOWNER	
	PATHSCHEMAS	SCHEMA	Applied to each schema
	TEXT	SCHEMA	Mask trigger name
		TGNAME	Mask tab/view/synonym
		OWNER	Mask UDT/UDF/STP
		TBNAME	
		SCHEMA	
		UDTNAME	
		UDFNAME	
		STPNAME	
		COLNAME	Mask column name
SYSVOLUMES	SGNAME	TSSGNAME	
	SGCREATOR	OWNER	
XSROBJECTS	XSROBJECTNAME	XMLSCHID	

Mask definition examples

Several examples of mask specifications are shown in the figures in this topic.

Note that some of the examples contain generic specifications, which are expressed by using an asterisk. The first mask that matches is used. The name is translated to the second value, or in the case where an attribute value is overwritten, the value of the attribute is overwritten to the new value.

```
Example 1:
NAME: ABC*,DEF*
```

In this example, any name that starts with ABC is changed to a name that starts with DEF in the generated SQL.

```
Example 2:
AUTHID: SYSIBM, COPY
```

In this example, all authids that have the value SYSIBM are translated to COPY.

```
Example 3:
TBNAME: *01*, *02*
```

In this example, a table that is named EMPLOYEE01 is translated to EMPLOYEE02.

```
Example 4:
DSSIZE: TESTDB.TESTTS*, REXX(PDDSSIZE,PARTITIONS,BPOOL)
```

In this example, the table spaces that start with TESTTS in the TESTDB database are changed to use the value that the REXX user exit PDDSSIZE returns as the DSSIZE.

Figure 527. Examples of translation masks

You can specify as many translation masks as you want. When a value is translated (for example, a name), the masks are processed one by one until a match is detected. A match means that the mask name is applicable to the value (for example, for a table name, mask names TBNAME and NAME are applicable) and the value conforms to the inputmask (for example, PRODTAB1 conforms to mask PROD*1). The value is translated based on the outputmask, or in the case where an attribute value is overwritten, the value of the attribute is overwritten to the new value. Only the first matching mask is used for a given value. If no matching mask is found, the value is not translated. Generally, you should put the most specific translation masks at the top of the mask file and the more general ones at the end.

```
Example 1:
COLNAME: COL*, NEWCOL*
```

In this example, any column name in any table that starts COL is changed to a column name that starts with NEWCOL. The column names that are changed include column names in triggers, views, and indexes. You cannot selectively change column names in specific tables.tart with TESTHRTS will be compressed.

```
Example 2:
COMPRESS: TESTDB.TESTTS*, YES
```

In this example, the table spaces in the TESTDB database that start with TESTTS will be compressed.

```
Example 3:
PRIQTY: TESTDB.*, 75%
```

In this example, the PRIQTY for all of the table spaces and index spaces in TESTDB database will changed to 75% of the current value of PRIQTY.

Figure 528. Examples of overwrite masks

```
Example 1:
TSBPNAME: TESTDB.TESTTS* : BP0,BP1
```

In this example, the bufferpool name BP0 is translated to BP1 for all of the table spaces in the TESTDB database that start with TESTTS.

```
Example 2:
VWSCHEMA: SCH*.VWA* : *, *TEST
```

In this example, view schemas for all views that have schema names starting with SCH and view names starting with VWA are changed to have TEST added to the end of the view schema names.

```
Example 3:
TSSGNAME: TESTDB : SG1,SG0
```

In this example, the storage group name for database TESTDB is changed from SG1 to SG0.

Figure 529. Examples of object-specific masking

Performance tip: Using many masks might increase processing time. If a match is not found early in the process, the program must search through the list of translation masks until a match is found.

Using verification masks

You can use verification masks to verify that the object attributes and names conform to the rules defined by the verification masks.

You might want to run verification requests on DB2 objects to ensure that your data adheres to certain rules. You can apply both existing masks and verification

masks at the same time. Verification masks are applied first, and then existing masks. All overwrite masks are allowed for verification masks.

You can use one of the following syntax formats to allow a VER request for an overwrite mask type:
- `VER,field:verification_operator,verification_values,RC=verification_rc`
- `VER,field:REXX(execname,parm1,...)`

The REXX exec must be built so that it returns two items in the return string. The first 2 items will be the return code value followed by a colon. The rest of the string is the error message if the REXX exec finds an error in verification.

You can refer to the following examples of VER masks:
- `VER,COMPRESS:EQ,Y,RC=4`
- `VER,PCTFREE:GT,20,RC=8`
- `VER,PCTFREE:RANGE,0,5,RC=4`

If any verification errors are encountered, the calling function will stop the process after all verification masks are applied. Verification errors are written to a separate DD file called VALOUT, which uses SYSOUT. If you are in TSO mode, use the `TSO ISRDDN` command to see the VALOUT file.

Note: The OBJNAME overwrite is only for verification masking and REXX syntax.

Chapter 23. Writing and modifying DB2 Admin applications

You can use DB2 Admin to create your own applications and tools using DB2 Admin, and you can extend existing applications.

The tasks are the same for both creating and extending applications.

Topics:

The application development process

DB2 Admin allows you to add new line commands to existing panels, and to develop new applications by using DB2 Admin as the dialog driver and interface to DB2.

Specifically:

- You can add new DB2 Admin functions to a copy of one or more of the panels supplied with the product.

  Tip: Use the existing code in the panel that you are modifying as a template, and make the necessary changes for the new function. When you complete your modifications, change the DB2 Admin source by creating an SMP/E usermod to ensure that changes are not lost if maintenance is applied to the product.

- You can develop new, independent applications by using the sample application panels included with DB2 Admin as templates.

Regardless of whether you are creating or extending DB2 Admin applications, the process involves creating ISPF panels that specify how DB2 Admin should perform SQL processing and dialog control.

Define your own line commands

You can define your own DB2 Admin line commands for each panel.

You might want to define commands that do the following actions:
- Start another ISPF-based tool with parameters from the current row
- Display the contents of related tables
- Change the contents of the displayed row by using an SQL statement

When encountering an unknown line command, DB2 Admin attempts to open an ISPF DB2 Admin line commands table with the same name as the panel that is

being displayed. If the table is found, DB2 Admin opens it and searches for the definition of the line command. If the line command is found, it is run.

Tip: The Tables, Views, and Aliases panel (ADB21T) can display multiple object types. The name of the line command table that is used for this panel depends on the object type that the line command is issued against. If you define your own line commands for panel ADB21T, be sure to read the comments in the EXEC about the different style that is used to defined the ISPF table.

Contents of the line command table

The DB2 Admin line command table contains the following columns:

CMD The line command. The line command must be the key in the table.

DESCR
> A description of the line command. This description is displayed if you enter a question mark (?) to request further information.

SQL The SQL statement that is run for this line command.

PAN The panel to be displayed as a result of this line command.

ISPF The ISPF statement that is run for this line command.

ACMD
> The DB2 Admin command that is run for this line command.

Creating a line command table

Create a line command table by writing a REXX EXEC that defines the ISPF table. A sample REXX EXEC, ADB21D, is provided in the SADBEXEC library. This EXEC provides a description of all possible line commands for the Database panel (ADB21D). It also defines four sample user-defined commands (USERI, USERS, USERD, and USERP). You can refer to this sample REXX EXEC when writing your own EXEC.

Some EXEC parts (such as ADB21T) use a different style to define the ISPF table than the style that is used in ADB21D EXEC. Be sure to read any comments in the EXEC in case the style that is used is different from the ADB21D sample REXX EXEC.

To enable line commands using your customized REXX EXEC:

1. Ensure that the REXX EXEC name (e.g., ADB21D) has a variable/value "table=ADB21D" that matches the DB2 Admin table display panel id (e.g., ADB21D).

2. On the ISPF command line, enter: TSO ALLOC F(ISPTABL) DA('<HLQ>.SADBTLIB') SHR REUSE.

3. Using Dialog Test ISPF option 7.6, enter: LIBDEF ISPTLIB DATASET ID("<HLQ>.SADBTLIB') STACK.

4. Update exec ADB21D with site-specific line commands and execute it by entering the command TSO EX "<HLQ>.SADBEXEC(ADB21D)'. This will create/update ISPF table "<HLQ>.SADBTLIB(ADB21D)', which the DB2 Admin Tool driver will use to display panel ADB20@ when the ? line command is entered on panel ADB21D.

Sample application

DB2 Admin includes a sample application that you can use to help you create your own applications.

The sample application consists of three ISPF panel source members located in library SADBPLIB. Their names are ADB2S, ADB2S1, and ADB2SU. Use these sample panels as templates to create your own application.

Recommendation: To better understand the concepts in this chapter, examine these ISPF panel source members.

The sample application shows how to maintain a small DB2 table called USER. The columns in the USER table are:

```
USERID      CHAR(08) NOT NULL
EMPNAME     CHAR(30) NOT NULL
EMPLNO      CHAR(05) NOT NULL
COMMENTS    CHAR(30) NOT NULL
```

Access the sample application from the DB2 Administration Menu panel by specifying option S (it is not included in the list of options). The DB2 Admin Sample Update Application panel, as shown in the following figure, is displayed.

```
DB2 Admin ------------- DB2 Admin Sample Update Application ------------- 01:14
Option ===>

   1 - Display/update the USER table                DB2 System: DB2X
   C - Create a USER table                          DB2 SQL ID: ISTJE
   I - Insert dummy entry into USER table
   D - Drop USER table
```

Figure 530. DB2 Admin Sample Update Application panel (ADB2S)

- Select option C on the Sample Update Application panel to create the *sqlid.USER* table (in default database DSNDB04).
- Select option I to insert a dummy row into the table so it is possible to display or update the table using option 1.
- Select option 1 to display the USER table. From this display, you can use line commands I, U, and D to insert, update, and delete rows.
- Select option D to drop the table.

Types of panels

You can create different types of panels with DB2 Admin.

The types of panels that you can create are:

Menu panels
These panels are typically at the top of a hierarchy of other panels. Menu panels specify the options that are available to the user.

Table display panels
These are ISPF table display panels on which data from DB2 or ISPF tables are displayed.

Data entry panels
On these panels, a user enters data that is input to a DB2 SQL statement, DB2 command, or DB2 Admin CLIST.

Help panels
These are standard ISPF help panels to guide the user in performing a task.

For a new application, you typically create a menu panel and a number of data entry and table display panels.

Controlling DB2 Admin processing

You control DB2 Admin processing by setting variables on the panels.

During processing, DB2 Admin looks at the variables and then processes the related commands or statements accordingly. If no variables are set, DB2 Admin redisplays the panel unchanged.

You can set the following variables on the panels:

PANEL
> The name of the next panel DB2 Admin should display. If this variable is used with an SQL SELECT statement, the next panel should be an ISPF table display panel that shows the rows returned by DB2. On a menu panel, set the PANEL variable to the panel name DB2 Admin should display for a particular choice.

SQLSTMT
> Any SQL statement that DB2 can execute. If the statement is an SQL SELECT, DB2 Admin creates an intermediate ISPF table, fetches the rows, adds the rows to the ISPF table, and shows the result on the specified panel. If no panel is specified, the default table display panel is shown. Multiple SQL statements can be specified; they must be separated by a semicolon (;).

ISPFSTMT
> Any ISPF statement that can be executed by the ISPEXEC ISPF API. This variable is useful for invoking your own CLISTs, EXECs, or other TSO/ISPF applications. Multiple statements can be specified; they must be separated by a semicolon (;).

DB2ACMD
> Any DB2 Admin primary command, which includes DB2 commands, ISPF statements, and SQL statements.

DB2 Admin processing flow

After a panel is displayed, DB2 Admin examines the variables and processes the instructions.

DB2 Admin examines the variables and processes the instructions according to the following rules:
- If the user presses END, the previous panel is displayed.
- If variable ISPFSTMT is set, all ISPF statements are processed first.
- If variable SQLSTMT is set, the SQL statements are processed one by one. If DB2 returns rows, the result on the panel named in the variable PANEL is displayed. If the variable PANEL is not set, the default panel is displayed.
- If the variable PANEL is set, the specified panel is displayed.
- If the variable DB2ACMD is set, the DB2 Admin commands are processed.

The process flow that DB2 Admin follows is shown in the following figure.
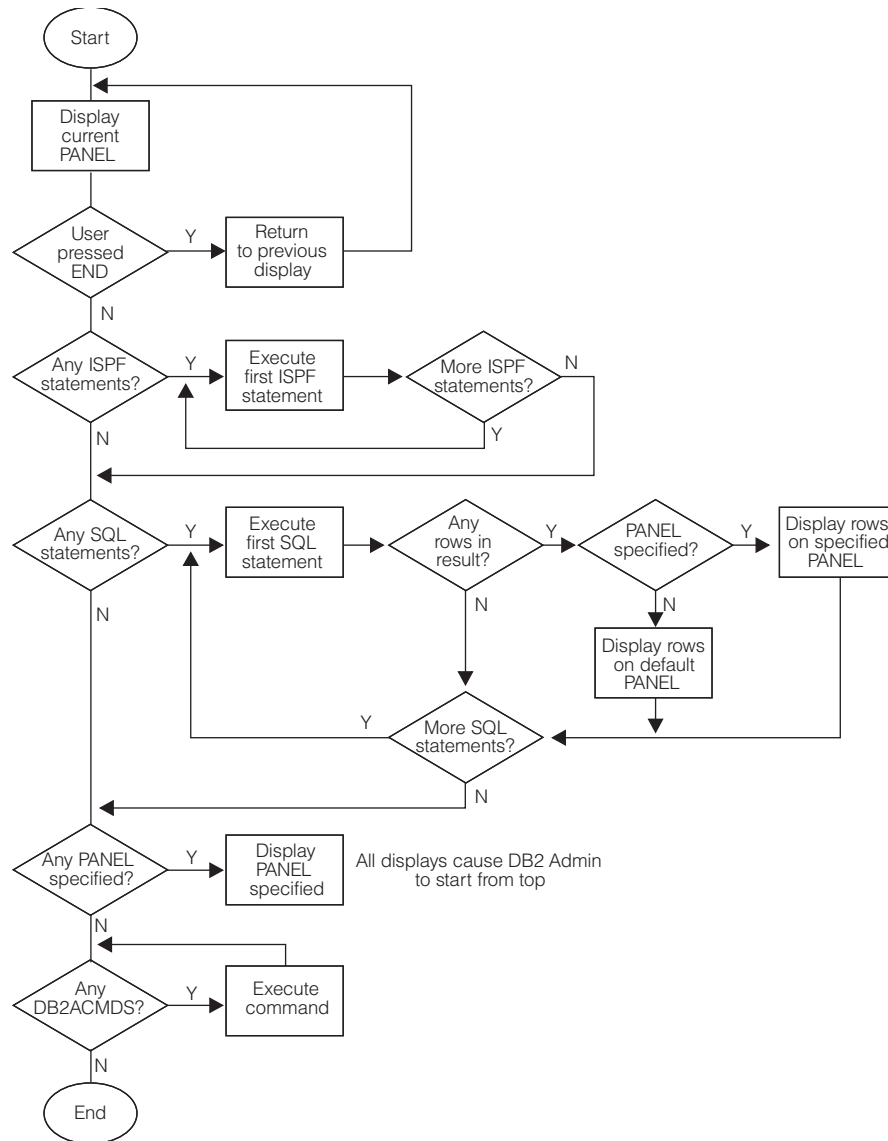
Figure 531. DB2 Admin logic diagram

Panel naming conventions

You can use DB2 Admin panels as a model to create your own panels.

However, you must use a different prefix in your panel names.

DB2 Admin panels have the prefix ADB. The suffix normally identifies the option that you selected to display the panel. For example, ADB1T is the panel for option 1 on the DB2 Administration Menu and option T on the following panel.

The corresponding help panels have the same name but use the prefix ADBH.

Using the DB2 Admin CLIST to invoke new applications

If you have created a new, independent application, you can use the DB2 Admin CLIST (ADBL) to invoke it.

Use the following parameters to invoke your application:

PANEL(panel)
Name of the first panel to be shown

SYSTEM(name)
DB2 subsystem that is to be used

Example: To start a DB2 Admin with your own customized panel, invoke the CLIST by issuing the following command:
```
%ADBL PANEL(yourpanel)
```

Updating rows using SQL

If your DB2 Admin application will use SQL to update rows, perform the updates on a separate panel.

Updating rows on the same panel will result in a copy of the data on the table display panel, but updated data in DB2. When you use a separate panel for updates, DB2 Admin refreshes the data in the table display panel automatically when DB2 data changes.

Also, DB2 Admin issues an SQL COMMIT before each display, so if you have concurrent users of your application, you probably should have a time stamp for the latest updates to rows.

If you are updating rows using SQL, consider using the structure shown in the following figure for your DB2 Admin application.
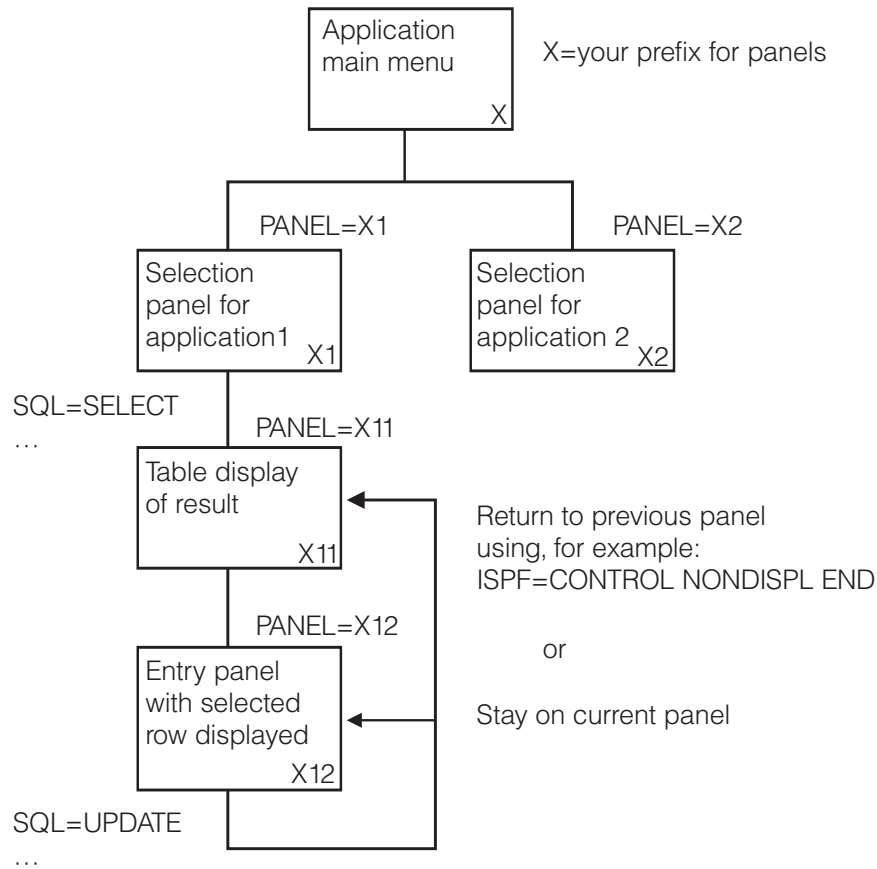
Figure 532. An example application structure

Using variables in your application

You can use two types of variables in your DB2 Admin application.

- General DB2 Admin variables
- Variables that contain column values, set as a result of an SQL SELECT and a line command that selected the row

All variables are located in the ISPF function pool.

General DB2 Admin variables

There are four general DB2 Admin variables: DB2SYS, DB2AUTH, MAXROWS, and DLEVEL.

The general DB2 Admin variables are:

DB2SYS
Indicates the DB2 system ID. The DB2SYS variable is set by the DB2 Admin CLIST.

DB2AUTH
Indicates the current DB2 authorization ID.

MAXROWS
> Indicates the maximum number of rows to fetch. The default is 1000.

DLEVEL
> Indicates the display level. The display level is increased by one for each nested display.

Variables that contain column values

After an SQL SELECT statement is executed, DB2 Admin defines a variable for each column of the result by using the ISPF VDEFINE service.

Therefore, these variables are available to your application. When you select a row, the content of the column variables have the values for that row.

The names of column variables are the same as DB2 column names except for the following differences:

- ISPF variable names have a maximum of eight characters. If the DB2 column name exceeds eight characters, it is truncated to eight characters. For example, the DB2 column name CLUSTERTYPE has the ISPF name CLUSTERT.
- Special characters, such as underscores in DB2 column names, are replaced by the at sign (@). For example, DB2 column name EMPL_NAME has the ISPF name EMPL@NAM.
- If duplicate column names exist in the result, all but the first duplicate column name are given ISPF name DUP0001, DUP0002, and so on. For example, SELECT CREATEDBAAUTH,CREATEDBCAUTH FROM SYSIBM.SYSUSERAUTH is given ISPF names CREATEDB and DUP0001.
- All DB2 SELECT expressions are given ISPF names COL0001, COL0002, and so on. For example, SELECT CURRENT DATE is given ISPF name COL0001.
- Table search argument variables are named in the same way as ISPF names, but they are truncated to seven characters and given the prefix @. Duplicates are named @DUP0001, @DUP0002, and so on.

Chapter 24. Using copies of the DB2 catalog

You can define copies of the DB2 catalog to DB2 Admin and create and maintain entries in the catalog copy version table, ADBCATVT.

One entry is required for each copy of the catalog being used.

Prerequisite: The catalog copy version table must already be created.

Topics:
- "Adding entries to the catalog copy version table"
- "Catalog copies at remote sites" on page 813
- "Using previously defined multiple copies of the DB2 catalog" on page 813

DB2 Admin enables you to use copies of the DB2 system catalog when selecting any of the options on the System Catalog panel. In addition, you can use the system catalog of a remote DB2 system.

For example, you can choose to use a different copy of the catalog for each weekday, and create a backup associated with each weekday. This strategy allows examination of previous definitions in the backup copies of the DB2 system catalog. Or you can allow only the system administrator to examine the active DB2 system catalog, and allow developers access to a copy of the DB2 system catalog. This strategy can result in decreased contention on the catalog caused by the developers' queries, while still allowing the system administrator to maintain the active DB2 system catalog.

Recommendation: When using multiple copies of the catalog, do not issue requests that involve data for which the definition of the objects has been changed since the catalog copy was refreshed.

DB2 Admin uses the catalog copy version table, ADBCATVT, to keep track of which DB2 copies are available to its users.

After the table is created, create an entry for each catalog copy to be used.

Adding entries to the catalog copy version table

You can add an entry to the catalog copy versions table (ADBCATVT).

About this task

To add an entry to the catalog copy version table:

Procedure

1. Select the CC option on the Administration Menu panel to display the Display Catalog Copy panel, as shown in the following figure. If no rows exist in the catalog copy versions table, the Insert an Entry panel is displayed instead, as shown in Figure 534 on page 811.

```
DB2 Admin ------------ DB2X Display Catalog Copy Versions --------- Row 1 of 6
Command ===>                                              Scroll ===> CSR

                                                   DB2 System: DB2X
Line commands:                                     DB2 SQL ID: ISTJE

 D - Delete  I - Insert  J - Create Copy, Bind Jobs

       Copy       Planname
Select Owner      Suffix   Timestamp               Type  Location
       *          *        *                       *     *
------ --------   -------- ----------------------- ----- ----------------
       V6ALI0     A6       ?                        A    STPLEX4A_DSN6
       V7COPY2    02       2001-07-16-13.57.16.2180 C
       V7COPY3    03       2001-07-16-16.34.55.7003 C
       V7COPY4    04       2002-04-04-16.56.19.5425 C
       V7COPY7    07       2003-04-11-16.33.37.6884 C
       V7NEW11    11       2003-04-14-17.21.05.2860 C
***************************** END OF DB2 DATA *****************************
```

Figure 533. Display Catalog Copy Versions panel (ADB2CCD)

The fields on this panel are:

Select   Input field where you enter one of the line commands listed on the
             panel. The supported line commands are:

   D   Delete a catalog copy entry from the table.

   I   Insert a new catalog copy entry into the table.

   J   Generate Create/Bind and Copy jobs to have DB2 Admin
               generate a job to create either the like tables or the aliases and
               bind the plans for that entry, and a job to copy the catalog.
               When you specify J, the Create Catalog Copy and Bind Batch
               Jobs panel (Figure 535 on page 812) is displayed so that you
               can enter additional information.

Highlevel qualifier
             Enter a valid DB2 authorization id. This field must be unique within
             the table.

Planname Suffix
             This can be any two characters. This field must be unique within the
             table.

Timestamp
             The time when the copy of the catalog was last refreshed. When
             inserting an entry, leave this field blank.

Type   The type of catalog:

   A   Indicates the entry is for a catalog at a remote site. When
               creating an entry for a remote catalog, enter the high-level
               qualifier, plan name suffix, type, and location of the remote
               catalog.

   C   Indicates the entry is for a copy of a local catalog. When
               creating an entry for a copy of a local catalog, enter the
               high-level qualifier, plan name suffix, and type.

   V   Indicates the entry is for views of a local catalog. When
               creating an entry for views of a local catalog, enter the
               high-level qualifier, plan name suffix, and type.

Location Name

Indicates the location of a remote catalog.

2. Issue the Insert line command to add an entry for each copy of the DB2 catalog that you want to use. The panel is shown in the following figure.

```
ADB2CCI n ---------------------- Insert an Entry ---------------------- 16:56
Command ===>

                                                   DB2 System: DSN9
Insert an entry into DB2 Catalog Copy Version Table    DB2 SQL ID: VNDMPM2

Enter/Verify:
  Copy Owner           . . .  VNDMPMG
  Plan Name Suffix     . . .  MF
  Timestamp            . . .  2009-07-29-11.20.45.586601
  Type                 . . .  C                  (C=copy, A=Alias, V=View)
  Location Name        . . .                  >  (Blank for types C and V)

Press ENTER to insert an entry, or press PF3 to cancel insert.
```

Figure 534. Insert an Entry panel (ADB2CCI)

Enter or verify the information in each field.

3. Press Enter to add the entry to ADVCATVT.

4. Issue the J command to generate create, bind, and copy jobs. The Create Catalog Copy and Bind Batch Jobs panel is displayed, as shown in the following figure.

   For Type A (aliases of a distributed DB2 system catalog), one job is created. **ALIBND***xx* (where *xx* is plan name suffix) creates aliases for the DB2 system catalog tables of the distributed subsystem at the given location. This job also binds the plans which DB2 Admin needs to access the aliases.

   For Type C (copies of a local DB2 system catalog), two jobs are created:

   - DDLBNDxx is the create and bind job. It creates the like tables for the copy and binds the plans. Run this job once to create all the tables for the copy of the catalog and to bind the plans that DB2 Admin is to use when this copy is selected.

   - CPYRUNxx is used to refresh the copy. Run it to create the initial copy of the tables; rerun it whenever the copy needs to be refreshed. The CPYRUNxx job also runs the RUNSTATS job against the table space that contains the copy and updates the timestamp field of the catalog copy version record.

   For Type V (views of a local DB2 system catalog), one job is created. **VIEBND***xx* (where *xx* is the plan name suffix) creates views for the local catalog tables. You can modify VIEBND*xx* to add predicates to the views to limit which rows are accessible to users. However, restricting which rows are accessible can effect the ability of DB2 Admin to retrieve information, resulting in incomplete information being returned.

```
ADB2CCJ n ---------DB2X Create Catalog Copy and Bind Batch Jobs -------- 16:56
Command ===>


Specify the following for DB2 Admin CATALOG COPY:        DB2 System: DSN9
                                                         DB2 SQL ID: VNDMPM2


Catalog Copy Owner . . : VNDMPMG   Type . . : C (C=copy, A=Alias, V=View)

Data set information:
  PDS for jobs  . . . . . . . .'VNDMPM2.TEST.JCL1'
  Prefix for work data sets . . VNDMPM2

Catalog Copy options (for type C only):
  Database Name   . . . . . . . VNDMPM2     (? to look up existing database)
  Storage Group Name  . . . . . VNDMPM2  >  (? to look up existing stogroup)
  Run SQLID . . . . . . . . . . VNDMPM2  >
  Catalog Copy Method . . . . . L            (L=LOAD from Cursor, U=UNLOAD/LOAD)
  Grant table privilege to  . . VNDM001
Miscellaneous options:
  Batch job PDS unit type . . . SYSDA
  Work data set unit type . . . SYSDA

Enter command BP to change batch job parameters
```

Figure 535. Create Catalog Copy and Bind Batch Jobs panel (ADB2CCJ)

5. Specify the name of an existing PDS where the generated jobs are to be stored. For type C (copies of a local DB2 system catalog), also specify:
   - The database and the storage group name to be used for the table space that will contain the like tables of the DB2 catalog tables. The name of the table space created to contain the like tables is the same as the qualifier of the copy.
   - The method to be used to copy the DB2 catalog to the like tables. The default is the LOAD from cursor method.

     Note: When you choose this option, the DB2 Administration Tool still uses the UNLOAD/LOAD method on catalog tables that contain LOB columns. This is because when a catalog table contains LOB columns, the catalog table also contains columns that are defined as GENERATED ALWAYS. DB2 does not allow GENERATED ALWAYS columns in the specification list when the LOAD from cursor method is used.

     The other method, UNLOAD/LOAD, allows you to unload into data sets as one process and to load as a second process. For the UNLOAD/LOAD method, the CPYRUN*xx* job will use TEMPLATE statements to define output and work data sets. Modify those TEMPLATE statements as necessary. (This function does not use any user-specified templates.)

     Note: Catalog Copy unloads LOB columns to a VBS data set if Admin Tool is running on DB2 V10 NFM.

     Recommendation: Use the LOAD from cursor method if the catalog data is not needed outside of the process, for example, for the movement or modification of data. The LOAD from cursor method reduces the I/O load of the entire process and requires no work data sets.
6. Press Enter to generate the jobs.
7. Run the appropriate job or jobs. The job only needs to be run once.
8. DB2 Admin renames any duplicate indexes that are created during CC processing. For the new names of the duplicate indexes, see step ISPFBAT.

Results

Recommendation: DB2 Admin performs space calculation of the catalog copy table space to build a reasonable CREATE TABLESPACE and CREATE INDEX statement. Run the RUNSTATS utility on the catalog table spaces before issuing the J line command on panel ADB2CJ. Verify that the space requirements are adequate.

Tip: The catalog copy process includes building tables that match the names of the system catalog tables. Also, indexes are built for those tables that match the names of the current set of indexes on the system catalog tables. Tables and indexes with the same qualifier and name might already exist as objects other than the intended catalog copy objects. If a duplicate object exists, SQLCODE -601 is issued when the DDL to create the new catalog copy is run. If you receive this error, you need to modify the DDL and restart the step. DB2 Admin detects certain duplicate index errors when creating the DDL for the index and attempts to avoid the error by creating a new name for the index that is based on the old name. However, DB2 Admin cannot detect and handle all cases.

Catalog copies at remote sites

Using multiple copies provides a method for using a remote site catalog that is different from the method provided by the DD (Distributed DB2 systems) option on the Admin main menu.

The GEN request is supported with the multiple copies method by using a catalog alias (Catalog Copy type 'A') and the alias' location for routing to the remote site.

Using previously defined multiple copies of the DB2 catalog

If your installation defined multiple copies of the DB2 catalog before you installed DB2 Admin, you need to perform an additional step after installing DB2 Admin.

About this task

Procedure

1. Reissue the J line command for each entry in the Display Catalog Copy Versions panel. Reissuing the J line command regenerates the jobs for the new release of DB2 Admin.
2. After the jobs are regenerated, run the BIND step of all DDLBNDxx and ALIBNDxx jobs.
3. Change the second line of the job from:

```
//*     RESTART=stepname,    <=== For restart, remove * and enter step
name
```

to

```
//      RESTART=BIND
```

Chapter 25. Running DB2 Admin across distributed systems

You can use DB2 Admin distributed support.

On remote systems, you can through the DB2 Admin Tool:

- Build utility jobs and submit them to run on remote systems.
- Perform alter and migrate functions for remote systems.
- Issue SQL statements against remote systems.
- Issue distributed GRANT and REVOKE commands.
- Issue other commands on remote systems.

To use DB2 Admin distributed support, select option DD from the DB2 Administration Menu panel to display the Distributed DB2 Systems panel, as shown in the following figure.

```
 DB2 Admin ----------------- Distributed DB2 Systems -------  ROW 1 TO 19 OF 19
 Command ===>                                          Scroll ===> PAGE

 Select the location you wish to use:                  DB2 System: DB2X
                                                       DB2 SQL ID: ISTJE
 Line commands:
    S - Use DDF to access remote catalog  CO - Connect to remote subsystem
  DIS - Display threads for remote system
Select Location
 ------ -----------------
        DENMARK_DB2M
        DENMARK_DB2X
        DENMARK_DB2D
        DENMARK_DB2T
        DENMARK_DB2W
        DENMARK_DB2P
        STOCKHLM_DB2B
        BELGHOLL_DB2
        OSLOMVSA_DB2T
        STOCKHLM_DB2C
        GER2_DSNS
        FINLAND_DB2
        LUBDB2
        NORDIC_DB2T
```

Figure 536. Distributed DB2 Systems panel (ADB2DDF)

This panel displays the remote DB2 subsystems that are available from the DB2 subsystem you are currently on (referred to as the *local subsystem*). Choose the DB2 subsystem for which you want the system catalog displayed. Press END to get back to the panel from which you came.

On the Distributed DB2 Systems panel, you can issue the following line commands:

DIS
  Displays the active threads for the location or system you select.

S Selects the remote subsystem for which you want to access the remote system catalog.

CO Connects you directly to a remote subsystem for issuing remote requests.

You can also use the CONNECT `location_name` primary command to connect to a remote subsystem.

Restrictions for connecting to a remote subsystem

- When using the distributed DB2 systems function to access a remote DB2 system catalog, some functions in the DB2 Admin system catalog dialog are disabled. For example, you cannot issue DB2 DISPLAY or GEN commands, and unless prompting is on, you also cannot issue DB2 BIND, REBIND, or FREE commands.
- If you connect to a remote subsystem that does not have an entry in the ADBTPARM customization table, then alter, migrate, and utility jobs are not allowed, and an error message is displayed. The DB2 subsystem parameters are stored in ISPF table member ADBTPARM, in the ISPTLIB table library that is specified in Tools Customizer by an administrator.
- To use copies of the system catalog of a remote subsystem, the local subsystem customization must specify the owner of the catalog copy version table.
- You cannot use option 1 of the Space Management function (display page set space by database).
- You cannot issue SM line commands on the database and table space panels.
- You cannot interface to other DB2 products from a remote subsystem.

Example: Accessing a remote subsystem

The following example shows you how to access a remote subsystem.

About this task

To access a remote subsystem:

Procedure

1. Enter S in front of the remote DB2 subsystem you want to access, as shown in the following figure.

```
 DB2 Admin ----------------- Distributed DB2 Systems -------  ROW 1 TO 19 OF 19
 Command ===>                                                Scroll ===> PAGE

 Select the location you wish to use:              DB2 System: DB2X
                                                   DB2 SQL ID: ISTJE
 Line commands:
   S - Use DDF to access remote catalog  CO - Connect to remote subsystem
 DIS - Display threads for remote system
Select Location
       *
 ------ ----------------
       DENMARK_DB2M
       DENMARK_DB2X
       DENMARK_DB2D
 S     DENMARK_DB2T
       DENMARK_DB2W
       DENMARK_DB2P
       STOCKHLM_DB2B
       BELGHOLL_DB2
       OSLOMVSA_DB2T
       STOCKHLM_DB2C
       GER2_DSNS
       FINLAND_DB2
       LUBDB2
       NORDIC_DB2T
```

Figure 537. Example of using distributed DB2 systems function (Part 1 of 2)

DB2 Admin displays the System Catalog panel, as shown in the following
figure, and indicates which location you are accessing. The release level and
mode of your DB2 subsystem affect the options that are available to you. All
generated batch utility jobs, ALTER commands, and MIGRATE commands are
sent to the remote subsystem (or the target system for the migrate jobs) for
execution after the jobs have been submitted on the local subsystem.

```
 DB2 Admin -------------------- DB2X System Catalog -------------------- 15:47
 Option ===>

 At location: DENMARK_DB2T                          DB2 System: DB2X
   AO - Authorization options                       DB2 SQL ID: ISTJE
    G - Storage groups           P - Plans
    D - Databases                L - Collections
    S - Table spaces             K - Packages
    T - Tables, views, and aliases  M - DBRMs
    V - Views                    H - Schemas
    A - Aliases                  E - User defined data types
    Y - Synonyms                 F - Functions
    X - Indexes                  O - Stored procedures
    C - Columns                  J - Triggers
    N - Constraints              Q - Sequences
   DS - Database structures     DSP - DS with plans and packages

 Enter standard selection criteria (Using a LIKE operator, criteria not saved):
 Name  . . . .                 > Grantor . . .         >
 Owner . . . .         >         Grantee . . .         >
 In D/L/H  . .                 >
 And/or other selection criteria (option xC shows you columns for option x)
 Column  . . .                 > Operator  . .      Value  . .
```

Figure 538. Example of using distributed DB2 systems function (Part 2 of 2)

2. Issue a BP command after connecting to the remote subsystem to set up JOB
   cards for the remote subsystem. The last JOB card that is used remains active
   until another BP command is issued. If you have not set up a JOB card for the
   remote subsystem, the JOB cards for the local subsystem are used on the
   remote subsystem.

Chapter 26. Troubleshooting

Use these topics to diagnose and correct problems that you experience with DB2 Admin.

Topics:
- "Gathering diagnostic information"
- "DB2 Admin messages and codes" on page 820

Gathering diagnostic information

Before you report a problem with DB2 Admin to IBM Software Support, you need to gather the appropriate diagnostic information.

If you receive DB2 Admin error messages that do not contain adequate information regarding the actions you should take, use the following information to diagnose common problems before contacting the IBM Support Center. The information that you gather to diagnose the problem is required when you open an incident with the DB2 Admin Support team.

- For general abends, obtain the following information:
  - ABEND code
  - Dump title
  - Failing module/CSECT name
  - A printout of the traceback from a Language Environment (LE) dump
  - Recent maintenance applied
  - Recent changes to the system
  - Frequency of abend, or prevailing conditions when the abend occurred. For example, does the abend occur for only a single user ID?
  - VTAM message
  - MVS ABENDs
  - Dumps, as appropriate
- Documentation that is required when contacting the support team:
  - DB2 Admin version number, release number, and maintenance level.
  - DB2 version number, release number, and maintenance level.
  - Is DB2 data sharing used?
  - Is a remote DB2 subsystem involved?
  - A complete explanation of the problem encountered.
  - Complete job output of failing jobs.
  - If problems occur using the ONLINE mode, send screen shots of any error messages and screen shots of all panels leading up to the error.
  - Appropriate input parameters for re-creating the problem scenario.
  - Complete DDL that fails, if appropriate.
  - A screen shot of the DB2 Admin Options panel.
  - Any work statement lists, mask data sets, or IGNORE data sets that apply.
- When troubleshooting the General Customization job ADBCUST with IBM, add the DEBUG=YES parameter as shown in the following figure. This parameter

produces trace information that can be shared and sent to IBM for further analysis.

```
ISFEPAN4     ADBCUSAX (J0032410) JCLEDIT                    Columns 00001 00072
Command ===>                                               Scroll ===> CSR
000095 //*  @END_CHANGE_HISTORY
000096 //**************************************************************************
000097 //*
000098 //ISPFBAT  EXEC PGM=IKJEFT01,REGION=0M
000099 //SYSEXEC  DD DISP=SHR,DSN=ADB.VA2FGRF1.EXEC
000100 //SYSTSPRT DD SYSOUT=*
000101 //SYSTSIN  DD *
000102 ISPSTART CMD( +
000103 %ADB2CUST SORT LISTPARM TCZCUST ADBCTLIB=RIVERAF.DEVCUST.ISPTLIB +
000104 DEBUG=YES)
000105 /*
000106 //SYSPRINT DD SYSOUT=*
000107 //ISPPROF  DD DISP=(NEW,DELETE,DELETE),
000108 //            DCB=(RECFM=FB,LRECL=80,BLKSIZE=7920,DSORG=PO),
000109 //            SPACE=(80,(1,5,10))
000110 //ISPLOG   DD SYSOUT=*,DCB=(LRECL=125,BLKSIZE=129,RECFM=VA)
000111 //ISPMLIB  DD DISP=SHR,DSN=SPF.PRODUCT.ISPMLIB
000112 //ISPPLIB  DD DISP=(NEW,DELETE,DELETE),
000113 //            DCB=(RECFM=FB,LRECL=80,DSORG=PO),SPACE=(80,(1,5,10))
000114 //ISPSLIB  DD DISP=(NEW,DELETE,DELETE),
000115 //            DCB=(RECFM=FB,LRECL=80,DSORG=PO),SPACE=(80,(1,5,10))
000116 //ISPTLIB  DD DISP=SHR,DSN=SPF.PRODUCT.ISPTLIB
000117 //VARS     DD *
```

Figure 539. General Customization job ADBCUST with DEBUG=YES parameter added

TSO ISRDDN

You might get a panel message that directs you to TSO ISRDDN.

If you receive the error message:

```
Analysis ended with return code = 12.  Use TSO ISRDDN to check the
file contents.  The files remain allocated and should be freed manually.
```

in a DB2 Admin panel, try the following procedures.

1. First try exit (PF3) out of the procedure and then try the procedure again. Sometimes exiting the procedure releases an existing process that is conflicting with the process you are trying to complete.
2. Next, access TSO and enter the command TSO ISRDDN. The command brings up a list of files that are currently allocated in the system. Review the file list to see if you can ascertain which file might be in conflict with your procedure.

DB2 Admin messages and codes

Use the information in these messages to help you diagnose and solve DB2 Admin problems.

Topics:
- "DB2 Admin Reverse Engineering condition codes"
- "DB2 Admin messages" on page 821

DB2 Admin Reverse Engineering condition codes

A DB2 Admin Reverse Engineering job that is running can issue condition codes.

The following condition codes can be issued:

0 Successful run.

4 Parameter error. The parameter is ignored, or the default is used. No generate requests are issued. Requested object is not found. A warning is issued.

8 No parameters found. Processing ended. The DB2 version is not yet supported. Other Errors might be issued.

12 DB2 version is not supported. Processing ended. Remote location is not defined or is not a DB2 MVS system. Internal error or limitation. Other severe errors are detected.

16 Severe error.

DB2 Admin messages

When you use DB2 Admin functionality, messages might be issued.

Not all DB2 Admin messages are included in this section.

ADB100E	**A parameter was omitted or an invalid parameter value was passed to module name** *module_name.*

Explanation: A required parameter has not been located in the parameter string passed to the program.

System action: None.

User response: If the parameter string was edited, provide the missing parameter, and ensure the parameter value is valid. If the parameter was omitted by the product, contact IBM Software Support.

ADB226E	**DB2 commands not available**

Explanation: The currently connected DB2 system does not support DB2 commands.

System action: Processing stops.

User response: Ensure that you are connected to a system that is running DB2 for z/OS and that it accepts DB2 commands.

ADB228E	**Invalid table name**

Explanation: The table name *table_name* is not allowed.

System action: Processing stops.

User response: Specify a valid table name and try the operation again.

ADB229E	**Panel error**

Explanation: An ISPF error occurred on display of panel *panel_name*, RC=*return_code*.

System action: Processing stops.

User response: Ensure that the specified panel is correct. If you are using the PANEL command ensure that the specified panel name is correct and that the panel can be used in this context. If the problem persists then contact IBM Software Support.

ADB230S	**No table displayed**

Explanation: The *command* command requires an active table to act on.

System action: Processing stops.

User response: Specify a valid table for the command and try the operation again.

ADB231E	**No table specified**

Explanation: The *command* command did not specify a table name.

System action: Processing stops.

User response: Specify a table name and try the operation again.

ADB232E	**Table error**

Explanation: An error occurred while processing ISPF table: *table_name*.

System action: Processing stops.

User response: Ensure that the table is a valid ISPF table.

ADB233E	**Invalid sort field**

Explanation: The specified sort field *field_name* is not present in the table.

System action: Processing stops.

User response: Specify a sort field that is in the table or use the sort command without parameters to display the fields that are in the table.

ADB249E	**Invalid data set**

Explanation: A command list data set must have DSORG=PO, RECFM=FB, and LRECL=80. The LISTDSI return code=*return_code*, and reason code=*reason_code*. LISTDSI reports that the data set has DSORG=*SYSDSORG*, RECFM=*SYSRECFM*, LRECL=*SYSLRECL*.'

System action: Processing stops.

User response: Specify a valid data set with the required attributes.

ADB252S   **Program Error**

Explanation: The display driver is stopped due to an internal error, oncode=*oncode*.

System action: Processing stops.

User response: Try the operation again. If the problem persists, contact IBM Software Support.

ADB259S   **DB2 V*version* unsupported**

Explanation: The version of DB2 that you are using is not supported by the version of DB2 Admin that you are using.

System action: Processing stops.

User response: Ensure that the version of DB2 Admin that you are using supports the version of DB2 that you are using.

ADB267I   *Operation* **was successful.**

Explanation: The *Operation* was successful. The SQL statement that was performed was a DB2 MERGE statement, so the target row was either updated or a new row was inserted.

Operation performed:

Update An existing row was located and updated.

Insert   An existing row was not located but a new row was inserted.

Explanation: Processing continues.

User response: None.

ADB268I   *Operation* **was successful. The originally specified row was not updated.**

Explanation: The *Operation* was successful. The SQL statement that was performed was a DB2 MERGE statement. The product detected that the user originated an action from one entry but changed the value used by DB2 MERGE to locate the row. This might lead to a new row being inserted or a different row being updated than was originally intended.

Operation performed:

Update An existing row was located and updated.

Insert   An existing row was not located but a new row was inserted.

Explanation: Processing continues.

User response: None.

ADB294E   **The unloading of LOB columns can require the use of templates. The templates can be generated only through work statement list (WSL) processing. Add the statements to a WSL and then run the WSL in batch mode to accomplish this task.**

Explanation: LOB columns must be unloaded to perform the operation. The unloading of LOB columns requires the use of templates, and templates can be generated only through work statement list (WSL) processing.

System action: Processing stops.

User response: Specify that the statements are to be added to a work statement list (WSL). When the WSL is run, ensure that it is run in batch mode.

ADB300E   **Module *module_name* DD statement is missing.**

Explanation: The specified DD statement is missing.

System action: Processing stops.

User response: Supply the missing DD statement and try again. Alternatively, regenerate the job and try again.

ADB318E   **The value must be *value_1*, or an integer between *value_2* and *value_3*.**

Explanation: The specified value is not allowed. The value must be equal to *value_1* or an integer between *value_2* and *value_3*.

System action: Processing stops.

User response: Enter a valid value and try the operation again.

ADB325E   **Invalid object type**

Explanation: The specified line command is invalid for the object type: *object_type*.

User response: Select a valid line command for this object type and try the operation again.

ADB331E   **Not possible.**

Explanation: This function is not possible when running against a Catalog Copy.

System action: Processing ends.

User response: No action is required.

ADB332I   **The string &db2aetok was found.**

Explanation: The requested string was found in the information displayed.

System action: Processing ends.

User response: None.

ADB332W   **The string &db2aetok was not found.**

Explanation: The requested string was not found in the information displayed.

System action: Processing ends.

User response: None.

ADB338E   **Invalid specification. Changing procedure types during CREATE is not allowed from this panel.**

Explanation: The following changes are not allowed when using the CRE (Create Like) line command from panel ADB21O:

- Changing from an external procedure, such as PLI, to an SQL procedure
- Changing from an SQL procedure to a non-SQL procedure
- Changing from an SQL external procedure to a native stored procedure

Changing from an external procedure to another of a different language is allowed, but not recommended.

System action: None.

User response: Restore the original language or native stored procedure value to the appropriate field.

ADB343E   **The specified keyword** *keyword* **can not be specified because** *reason*.

Explanation: The specified keyword is not valid because of the reasons listed below. If keyword is PARALLEL for the LOAD utility, there are two reasons:

1. The table to be loaded has LOB or XML columns and SHRLEVEL NONE is specified.
2. The table to be loaded has XML columns and is in a simple or segmented table space and SHRLEVEL CHANGE is specified.

System action: Processing stops.

User response: If keyword is PARALLEL, specify a valid keyword and try the operation again.

ADB359E   **The LC line command cannot be used on a view that is defined on more than one table. Use the T line command to locate the table that you want to process.**

Explanation: The view selected is defined on more than one table but the LC line command specified can only operate on a single table. The line command cannot be executed because the target of the load is ambiguous.

System action: Processing stops.

User response: Use the T line command to display the tables associated with the view. Locate the table that you want to process and then issue the U.LC line command against the specific table.

ADB362E   **Enter string**

Explanation: A character string was not specified in the FIND command.

System action: Processing stops.

User response: Enter the string of characters to be found.

ADB363E   **Invalid string**

Explanation: The FIND string cannot be a null ("") string.

System action: Processing stops.

User response: Specify a non-null string of characters to search for in the FIND command.

ADB364E   **Invalid column number**

Explanation: The column number in the FIND command is invalid.

System action: Processing stops.

User response: Specify a valid column number and issue the FIND command again.

ADB365E   **FROM column > TO column**

Explanation: The FROM column that was specified in a FIND command is greater than the TO column that was specified in the command.

System action: Processing stops.

User response: Specify a FROM column number that is less than the TO column number and issue the FIND command again.

ADB366E   **Enter a FIND command**

Explanation: The RFIND command is used to reissue the FIND command that was previously issued.

System action: Processing stops.

User response: Issue a FIND command first then issue the RFIND command.

---

an A or D line command on the altered TB or altered TS which has a dependency relationship, the table space needed by the altered table might not be created, which can lead to an error when you apply changes later.

System action: Processing continues.

User response: No action is required.

ADB456E **The database already exists. Enter a new database name.**

Explanation: The database cannot be renamed to an existing database name.

System action: None.

User response: Enter a new database name in the **New database name** field and press Enter.

ADB461E **A system-managed table must have columns defined as ROW BEGIN and ROW END for the PERIOD clause. Either one or both columns of this type are missing in this table.**

Explanation: A request for a System period has been made without valid columns for the start and end columns of the period in the table definition.

System action: None.

User response: Return to the column definition panel and assure that there are columns with the ROW BEGIN and ROW END attributes defined before proceeding.

ADB462E **Specify both a start and an end column.**

Explanation: You must specify both a start and end column for the BUSINESS_TIME period on the Select BUSINESS TIME Period Columns panel .

System action: None.

User response: Use the S and E line commands to select the Start and End columns for the BUSINESS_TIME period. Use CANCEL to return to the Create Table Columns panel without making a selection. .

ADB463E **Only one start and one end column are allowed.**

Explanation: You cannot specify more than one start and one end column for the BUSINESS period.

System action: None.

User response: Use the R command to remove any duplicate selection.

ADB464E **You must have at least two TIMESTAMP(6) WITHOUT TIME ZONE or two DATE columns valid for BUSINESS_TIME period columns before proceeding.**

Explanation: There must be at least two columns which are valid for the business period start and end columns before proceeding to the Select BUSINESS TIME Period Columns panel.

System action: None.

User response: Add or redefine columns on the Create Table Columns panel to assure that there are two columns valid for the business period.

ADB465E **A request for BUSINESS_TIME WITHOUT OVERLAPS for the constraint without a BUSINESS_TIME period will be ignored.**

Explanation: This request will be ignored if you specify YES for the BUSINESS_TIME WITHOUT OVERLAPS option when defining a primary key, if you have not already defined a BUSINESS_TIME period. If you do not define a BUSINESS_TIME period before issuing the CREATE command, the option will be ignored.

System action: None.

User response: No action is required. If you want to use the option, define a BUSINESS_TIME period.

ADB466E **The BUSINESS_TIME WITHOUT OVERLAPS option is invalid because a BUSINESS_TIME period start or end column matches a column in the primary key.**

Explanation: The BUSINESS_TIME WITHOUT OVERLAPS option is not valid if a start or end column of the business period matches any of the keys of the primary constraint.

System action: None.

User response: Either change the business period start or end column, or change the primary key columns so that they do not conflict.

ADB467E **There might be some options from the model table which are not used.**

Explanation: MODEL=YES was specified from the main Create Table panel. Certain options might not be adopted from the model table.

System action: None.

User response: No action is required.

ADB468E   **Use the TBLOPTS command to specify a SYSTEM_TIME period.**

Explanation: Columns with attributes AS ROW BEGIN and AS ROW END have been specified.

System action: None.

User response: Go to the Create Table Options panel to specify a SYSTEM_TIME period. .

ADB471E   **The specified database name is implicit. Enter a new database name.**

Explanation: The database cannot be renamed to an implicit database name, such as DSN*nnnnn* where *nnnnn* is a numeric value.

System action: None.

User response: Enter a new database name in the **New database name** field and press Enter.

ADB472W   **No utilities (except UNLOAD) will be generated for implicit table spaces.**

Explanation: The RENDB function will not generate utilities for implicit table spaces.

System action: This warning message is displayed if the database to be renamed has at least one implicit table space.

User response: Press Enter if you want to continue processing.

ADB473E   **The specified database name is reserved. Enter a new database name.**

Explanation: The database cannot be renamed to a reserved database name of DSNDB01, DSNDB04, DSNDB06, or DSNDB07.

System action: None.

User response: Enter a new database name in the **New database name** field and press Enter.

ADB526E   **An XML column defined as NOT NULL and no default cannot be added.**

Explanation: An XML column cannot be added with the NOT NULL attribute and no default, since there is no default data value to LOAD for columns.

System action: Processing stops.

User response: Re-specify the attributes to allow null values.

ADB539E   **The target SSID *DB2_SSID* cannot be found in customization table. Ensure that the SSID customization table is properly defined.**

Explanation: The SSID for the target DB2® subsystem cannot be found.

System action: Processing stops.

User response: Ensure that the SSID is defined in the ADBTPARM member. Using Tools Customizer, edit the SSID, generate the customization jobs, and submit the ADBCUST job that corresponds to the SSID that you edited. When the ADBCUST job is submitted, the SSID will be added to the ADBTPARM member.

ADB559F   **The LOAD job member names to be generated exceed eight characters. Specify a prefix that is less than five characters for the job member names.**

Explanation: Because numerous tables are being processed, the LOAD job member names to be generated exceed eight characters (ADBTSnnRL) This error occurs when the table space being Altered or Redefined has more than nine tables and the following options are specified:

> Combine job steps=NO
>
> Member name or prefix=ADBTS (five chars)
>
> Unload Method=H

System action: Processing stops.

User response: Specify a prefix that is less than five characters for the job member names.

ADB559G   **YES is not allowed when moving to a Partitioned-by-Growth table space.**

Explanation: It is not permitted to redefine a table space to Partitioned-by-Growth or Partitioned-by-Range with the Member Cluster input field set to YES. This is a DB2 V9 restriction.

System action: Processing stops.

User response: Specify NO in the Member Cluster input field when you redefine a Table Space to Partitioned-by-Growth or Partitioned-by-Range.

ADB559O   ***creator.name* contains *n* tables. Converting to a partitioned table space is not supported, therefore options for partitions cannot be changed.**

Explanation: The database cannot be partitioned because it contains more than one table.

System action: Processing stops.

User response: You can continue with other line commands or press PF3 to leave the panel.

ADB559P   **Only converting to a Partition-by-Growth (PBG) or a Partition-by-Range (PBR) table space is permitted.**

Explanation: The table space is PBG or PBR, but the number of partitions or the segment size cannot be changed.

System action: Processing stops.

User response: Enter ORIGINAL on the command line to reset the values to the original values.

ADB559Q   **A table space name is required when moving to *type*.**

Explanation: The number of partitions or the segment size was changed, but no table space name was provided. The *type* can be:

- Partitioned-by-Growth table space (PBG)
- Partitioned-by-Range table space (PBR)
- Partitioned table space

System action: Processing stops.

User response: Provide a table space name, or type ORIGINAL on the command line to reset the values to the original values.

ADB586E   **An UNLDDN template is required when SPANNED=YES is set.**

Explanation: An UNLDDN template must be provided in order for DB2 to determine the space necessary and to create an unload file with the spanned attribute.

System action: Processing stops.

User response: On panel ADB28M, specify the TU option. On panel ADB25TU3, specify UTILITY TEMPLATE Usage, and define a template for UNLDDN. Return to panel ADB28M to set the 'Generate template statements' field to YES, then press Enter.

Alternatively, change the 'Spanned' field to NO.

ADB587E   **The HIDDEN attribute is not allowed for a column defined as ROWID.**

Explanation: If a column is defined with a ROWID data type, then the column cannot be specified as HIDDEN.

System action: Processing stops.

User response: Either change the HIDDEN attribute to NO, or specify a different column type.

ADB588E   **You must change one or more keys in order to change the primary key constraint name.**

Explanation: You cannot change the primary key constraint name without also changing one or more keys for the constraint.

System action: Processing stops.

User response: Change one or more of the constraint columns, or restore the original constraint name.

ADB589E   **You must choose one or more columns for the constraint key.**

Explanation: You must specify one or more columns for the constraint key when adding a primary or unique key constraint.

System action: Processing stops.

User response: Specify one or more columns for the constraint key before proceeding.

ADB600E   **Invalid time. The specified value must be formatted as (+/-)hh:mm. The hh parameter must be between -12 and +14 and mm between 00 and 59.**

Explanation: The time value is not specified in the correct format. The value must be formatted as (+/-)hh:mm. The hh parameter must be a numeric value between -12 and +14 and the mm parameter must be a number between 00 and 59.

System action: Processing stops.

User response: Specify the time value using valid formatting and try the operation again.

ADB614I   **The Real-Time Statistics for the object have been refreshed.**

Explanation: The REFRTS command completed successfully and the real-time statistics have been updated.

System action: Processing continues.

User response: No action is required.

ADB614E   **The CCSID values must match.**

Explanation: The encoding scheme specified for the array subtype must match the encoding scheme for the source type.

System action: Processing ends.

User response: Change the encoding scheme inputs so that they match and try the operation again.

ADB615E   Invalid length value.

Explanation: Do not specify length with *array* array subtype. Length can be specified only for VARCHAR array subtype.

System action: Processing ends.

User response: Remove the length value if you are using an INTEGER array subtype and try the operation again.

ADB616E   Invalid CCSID option.

Explanation: *array* is a valid array subtype. CCSID can be specified only for VARCHAR array subtype.

System action: Processing ends.

User response: Remove the CCSID value input or change the array subtype and try the operation again.

ADB617E   Invalid data subtype.

Explanation: *type* data subtype is invalid with *array* array subtype.

System action: Processing ends.

User response: Change the data subtype and array subtype to values that are valid and try the operation again.

ADB618E   Invalid data type.

Explanation: The user data type specified is invalid for the CREATE *procedure/function* statement that is being built.

System action: Processing ends.

User response: Change the data type to a value that is valid and try the operation again.

ADB700E   Column not allowed. Column
_column_name_ cannot be specified as part
of the primary key because it is a
DECFLOAT, XML, or LOB data type, or
it is a row-change-timestamp column.

Explanation: DB2 does not allow a column of the indicated type to be included as one of the primary key columns for the table.

System action: Processing stops.

User response: Remove the column from the primary key specification.

ADB701E   Column not allowed. Column
_column_name_ cannot be specified as part
of the unique key because it is a LOB
data type.

Explanation: DB2 does not allow a unique key to be

created for a LOB column data type.

System action: Processing stops.

User response: Remove the column from the specification.

ADB702E   Column not allowed. Column
_column_name_ cannot be specified as part
of the foreign key because it is a LOB
data type.

Explanation: DB2 does not allow a column of the indicated type to be included in a foreign key definition.

System action: Processing stops.

User response: Remove the column from the specification.

ADB703E   Column not allowed. Column
_column_name_ cannot be specified as a
column of a parent key in a
REFERENCES clause because it is a
LOB data type.

Explanation: DB2 does not allow a column of the indicated type to be included in a REFERENCES clause.

System action: Processing stops.

User response: Remove the column from the specification.

ADB704E   Column not allowed. Column
_column_name_ cannot be specified as a
column of a partitioning key because it
is a _data_type_ data type.

Explanation: DB2 does not allow a column of the indicated type to be included as one of the partitioning columns for the table.

System action: Processing stops.

User response: Remove the column from the specification.

ADB705E   Operation not allowed. Column
_column_name_ cannot be changed to a
LOB column because a check constraint
exists on this column.

Explanation: DB2 does not allow a column of the indicated type to be included in a check constraint.

System action: Processing stops.

User response: Remove the column from the specification.

ADB706E   **Operation not allowed. Column** *column_name* **cannot be changed to a LOB column because a field procedure exists on this column.**

Explanation: A column with a field procedure cannot be changed to a LOB data type.

System action: Processing stops.

User response: Remove the field procedure prior to changing the column definition.

ADB707E   **Operation not allowed. Column** *column_name* **cannot have a default value. Only NULL is allowed.**

Explanation: DB2 does not allow the column to have a default value. Specifying NULL is allowed.

System action: Processing stops.

User response: Specify NULL as required.

ADB708E   **Operation not allowed. Column** *column_name* **cannot be converted from a LOB data type to any other data type.**

Explanation: Data type conversion from a LOB data type is not allowed.

System action: Processing stops.

User response: Specify a data type conversion that is allowed.

ADB709E   **Column not allowed. Column** *column_name* **cannot be provided as a column in the constraint because it is a DECFLOAT, XML or LOB data type, or it is a row-change- timestamp column.**

Explanation: DB2 does not allow a column of the indicated type to be included in the constraint

System action: Processing stops.

User response: Remove the column from the specification.

ADB710E   **Operation not allowed. Column** *column_name* **cannot be converted from NULL to NOT NULL.**

Explanation: The column cannot be converted from NULL to NOT NULL.

System action: Processing stops.

User response: Retain the NULL specification.

ADB711E   **This operation is not allowed against a hidden column.**

Explanation: The line command that you entered is not allowed on a hidden column.

System action: Processing stops.

User response: Do not issue the command against the column.

ADB712E   **Improper length. A LOB column cannot be shortened in length.**

Explanation: A LOB column's length cannot be reduced.

System action: Processing stops.

User response: Retain the original column's length.

ADB720E   **Column** *column_name* **cannot be specified as a column of an index key due to its data type,** *data_type***.**

Explanation: A column of the selected data type cannot be specified as part of an index.

System action: Processing stops.

User response: Select a column with a data type that can be part of an index.

ADB722E   **The RECLUSTER option is not allowed with option SORTDATA specified as YES or BLANK.**

Explanation: If the SORTDATA option is specified as YES or BLANK, you cannot specify the RECLUSTER option.

System action: Processing stops.

User response: Specify SORTDATA NO with the RECLUSTER option.

ADB723E   **Operation not allowed. A table defined with DATA CAPTURE CHANGES cannot be placed into a NOT LOGGED table space.**

Explanation: A table defined with the DATA CAPTURE CHANGES attribute cannot be placed into a table space defined with the NOT LOGGED attribute.

System action: Processing stops.

User response: Specify a table space with the proper DB2 logging attribute, or remove the DATA CAPTURE CHANGES attribute from the table.

ADB724E **Operation not allowed. A table cannot be moved to an implicitly created database or table space.**

Explanation: A table cannot be placed into a table space which was implicitly created by DB2. .

System action: Processing stops.

User response: Specify a table space that was explicitly created.

ADB725E **A row change timestamp column cannot be added to the table.**

Explanation: Adding a ROW CHANGE TIMESTAMP column is not permitted.

System action: Processing stops.

User response: Respecify the column without the ROW CHANGE TIMESTAMP attribute.

ADB726E **Conversion to or from a row change timestamp column is not allowed.**

Explanation: Changing to or from a ROW CHANGE TIMESTAMP column is not permitted

System action: Processing stops.

User response: Respecify the column without changing to or from a ROW CHANGE TIMESTAMP attribute.

ADB727W **Different columns in the primary key definition were specified, or the length of a primary key column was increased.**

Explanation: The primary key columns for the table were respecified, or a column length was changed. The primary key must be dropped first, which will result in the loss of any referential integrity definition based upon the primary key definition.

System action: This message is issued as a warning and an action prompt panel is displayed.

User response: An ALTER TABLE ... DROP PRIMARY KEY statement is needed to perform this change. Use option 3 to create new referential constraints and any required new indexes.

ADB728E **Conversion from** *column_type* **to** *new_column_type* **data type, or changing the length of a** *column_type* **data type is not allowed.**

Explanation: Changing the data type to or from the indicated data type is not permitted.

System action: Processing stops.

User response: Respecify the column without changing the data type.

ADB729E **Conversion from** *column_type* **to** *new_column_type* **data type is not allowed since the source column is not defined as FOR BIT DATA.**

Explanation: The original column is not defined as FOR BIT DATA. Conversion is only allowed on FOR BIT DATA columns.

System action: Processing stops.

User response: None.

ADB730E **Operation not allowed. The target table space must be of the same partitioning type as the existing table space (partition by growth or partition by range).**

Explanation: Moving a table to a table space of a different format when the old or new table space is partitioned by growth is not permitted.

System action: Processing stops.

User response: Specify a target table space of the same type as the table's current table space.

ADB731E **Too many operations performed. Only one operation is allowed at a time.**

Explanation: The combination of operations is not allowed.

System action: Processing stops.

User response: Specify one operation at a time.

ADB731E **Too many operations performed. Only one operation is allowed at a time.**

Explanation: The combination of operations is not allowed.

System action: Processing stops.

User response: Specify one operation at a time.

ADB735E **An upgrade cannot be done. The table** *table_name* **can only be upgraded from the previous release to the current release. Re-create the table.**

Explanation: An upgrade cannot be done to the control table *table_name* because it is not at the proper level.

System action: None.

User response: Drop and re-create the TEMPLATE control table.

ADB737E   **Incorrect table format. The table** *table_name* **does not have the expected column names, data types, or both. Check the current definition of the TEMPLATE control table.**

Explanation: The identified TEMPLATE control table cannot be upgraded because the table definition is incorrect.

System action: None.

User response: Check the table name and the table owner to see if it is a control table. LISTDEF and TEMPLATE control tables are DB2 control tables. Thus, they could be created during DB2 installation by the DSNTIJCC member. DB2 Administration Tool could also be used to create LISTDEF and TEMPLATE control tables. The default name for LISTDEF control tables is DSNACC.UTLIST, and the default name for TEMPLATE control tables is DSNACC.UTTEMPLATE. See LISTDEFs and TEMPLATEs in this User Guide for further information.

ADB748E   **There has been an unsupported request** *request_type* **for exec ADBEUSV.**

Explanation: There might be a mismatch between panel ADB2USV and exec ADBEUSV.

System action: Processing stops.

User response:
1. Log off, log on, and try the procedure again.
2. If the problem persists, contact IBM software support

ADB79AW   **A unique key constraint named,** *constraint_name* **already exists for this table. You can change the constraint name, enter CONTINUE to replace the keys for the constraint, or END to exit.**

Explanation: The table already has a unique key constraint with this name, or a constraint was added within this ALT session.

System action: This is a warning message.

User response: Try these actions to correct the problem:
- Change the constraint name
- Enter CONTINUE to replace the key, or END to exit without saving.

ADB799E   **The table space is not a range partitioned table space.**

Explanation: The LKEY line command was issued, but it is not valid for partition by growth table spaces. This line command is only valid for range partitioned table spaces.

System action: The system waits for the next command.

User response: Issue a different command and press Enter or press PF3 to leave the panel.

ADB799W   **A primary key constraint already exists for this table. Enter CONTINUE to replace the key, or END to exit.**

Explanation: The table already has a primary key, or one was added within this ALT session. Only one primary key is allowed.

System action: This is a warning message.

User response: Enter CONTINUE to replace the key, or END to exit without saving.

ADB811E   **NO is not valid for this option because Drop Impact Report is specified as YES or BATCH.**

Explanation: If you set the **Show this panel prior to each drop** field to NO, then you cannot set the **Display Drop Impact Report** field to YES or BATCH. The settings are not compatible.

System action: Processing stops.

User response: Set both the **Display Drop Impact Report** field and **Show this panel prior to each drop** field to NO. Alternatively, specify YES or blank in the **Show this panel prior to each drop** field and specify YES in the **Display Drop Impact Report** field.

ADB812E   **Lines that are marked with '?' are not committed to change. Remove the '?' and press Enter to commit the change.**

Explanation: A value for an object was marked to change but the change is not committed by pressing Enter once. You must press Enter again.

System action: DB2 Admin puts a question mark in the line command field and puts the statement 'modify pending' in the message column.

User response: Remove the question mark from the lines that you want modified and press Enter to continue.

ADB815E   **This table is not an archive enabled table.**

Explanation: The ARCH line command was issued for a table that is not archive enabled. The command cannot be processed.

System action: The system waits for the next user action.

User response: Issue the ARCH line command for table objects that have been archive enabled. Use the BROWSE primary command from panel ADB21T to see

catalog information from SYSTABLES. Archive enabled tables are those with TYPE=T and with the ARCHIVING_SCHEMA ARCHIVING_TABLE columns having the schema and name of the archive table.

ADB900E   **Error condition. An unrecognized object type** *object_type* **was passed when virtual changes were applied.**

Explanation: The object type is unrecognized. It is unlikely that this error will cause a problem.

System action: Processing continues.

User response: Contact IBM support to report the message.

ADB901E   **An error occurred in the** *program_name*. **Return code =** *return_code*.

Explanation: An error occurred in the specified program. The program cannot continue.

System action: Processing stops.

User response: Contact IBM support to report the message.

ADB903I   **The pending definition changes have been dropped.**

Explanation: The pending DB2 definition changes have been dropped from the SYSPENDINGDDL table.

System action: Processing continues.

User response: No action is required.

ADB904E   **The table** *table_name* **contains too many columns.**

Explanation: You can assign up to 750 columns for a non-dependent table. Dependent tables can have up to 749 columns.

System action: Processing stops.

User response: Limit the number of columns to allowed values and try the operation again.

ADB906E   **Export changes failed. Use TSO ISRDDN to view the ADBDIAG file contents and determine the cause of failure.**

Explanation: The export changes procedure failed. Use TSO ISRDDN to check the ADBDIAG file contents. In the ADBDIAG file, you might find references to objects involved in the failed export changes procedure.

System action: Processing stops.

User response: Use TSO ISRDDN to check the ADBDIAG file contents. Review objects or messages in the file that indicate conflict.

ADB907E   **The primary command is invalid. The valid primary command is** *&validcmd.*

Explanation: To add a product entry, use the primary command ADD. To update a product entry, use the primary command UPDATE or UPD. To delete a product entry, use primary command DELETE or DEL.

System action: Processing stops.

User response: Enter a valid value for the primary command.

ADB908E   **Invalid buffer pool size. The buffer pool must be** *&bpm.* **and the size cannot be altered. To alter the buffer pool size to something other than &bpm, enter END to exit and return to the Table Space panel (ADB21S). Then, use the line command ALT to redefine the table space. Do not use the AL command to change the buffer pool size to a different buffer pool size.**

Explanation: The buffer pool size must be appropriate for the table space. If the buffer pool size of the table space is 4KB, the value of *&bpm.* is BP0-BP49, 8KB is BP8K0-BP8K9, 16KB is BP16K0-BP16K9, and 32KB is BP32K, BP32K1-BP32K9.

System action: Processing stops.

User response: Use the line command ALT to redefine the table space. Do not use the AL line command.

ADB909E   **The Installation default parameters option is not available because Change Management was disabled at install time.**

Explanation: The **Installation default parameters** option is not available because the Change Management database was not created or the CM option was disabled at install time. DB2 Admin will use DB2 Utility default values instead.

System action: Processing stops. The DB2 Admin utility panels will allow you to specify the PARALLEL parameter according to the standard DB2 utility limits. See the DB2 Utility Guide and Reference for more information about the PARALLEL keyword.

User response: If the DB2 Utility default value limits are sufficient, then no action is needed.

If there is a need to enable the **Change installation default parameters** option on the DB2 Admin Options panel (ADB2P), the DB2 Admin administrator or installer should complete the following steps.

1. In Tools Customizer, navigate to the Customizer workplace: DB2 Admin Tool panel (CCQPWRK).
2. Issue the E line command for the Product parameters field.

3. On the Product parameters panel (CCQPPRD), scroll several pages to the Admin Tool setup task (create and upgrade) section, and enable the following options:

   • Change Management database - YES

   • Enable CM on DB2 Admin primary menu - YES

4. Press PF3 to navigate back to the Customizer Workplace: DB2 Admin Tool panel (CCQPWRK).

5. Issue the G line command to regenerate the Admin Tool Setup Task job template ADBSETUP.

6. Submit the Admin Tool Setup Task job template ADBSETUP.

7. Submit the ADBBIND template.

ADB986E   **Incorrect <parameter1> specified for default "<parameter2>". Use the U command to specify a valid <parameter1>.**

Explanation: An incorrect special register or session variable name has been specified for the default value.

System action: Processing stops.

User response: Either correct the default value, or use the U (update) command to change the special register or session variable with the GENERATED option.

ADB987E   **The default value <parameter1> is valid only on a column with <parameter2> data type.**

Explanation: You have specified an invalid data type or length for the data change operation, special register, or session variable used in the GENERATED expression.

System action: Processing stops.

User response: Correct the data type and /or length.

ADB988E   **A special register or session variable must be specified. Use the U (update) command to specify one using the GENERATED option.**

Explanation: A special register or session variable is missing for the GENERATED clause, as indicated by a default value of "a" or "b".

System action: Processing stops.

User response: Use the U (update) command to specify a special register or session variable with the GENERATED option.

ADB991E   **The archive table cannot be defined as a parent or child in a referential constraint.**

Explanation: You cannot specify an archive table that

is defined as a parent or child in an existing referential constraint.

System action: Processing stops.

User response: Specify an archive table that is not defined as the parent or child in an existing referential constraint.

ADB992E   **The archive-enabled table and the archive table must have the same <parameter>.**

Explanation: The archive-enabled table and it's archive table must have the same encoding scheme and number of columns.

System action: Processing stops.

User response: Specify an archive table that has the same number of columns and the same encoding scheme as the archive-enabled table.

ADB993E   **The <parameter> table must be the only table in the table space.**

Explanation: In order to enable archiving, the specified table must be the only table in the table space.

System action: Processing stops.

User response: Specify a table that is the only table in the table space.

ADB994E   **The <parameter> cannot include a SYSTEM_TIME or BUSINESS_TIME period.**

Explanation: An archive-enabled or archive table cannot include a SYSTEM or BUSINESS time period.

System action: Processing stops.

User response: Specify a table that does not contain a period.

ADB995E   **The <parameter1> table cannot include <parameter2>.**

Explanation: In order to enable archiving, neither the archive-enabled table nor the archive table can include any of the following:

• An identity, transaction-start-ID, row-begin, or row-end column

• A column mask or row permission

System action: Processing stops.

User response: Assure the archive and archive-enabled tables do not contain any of the above column attributes.

ADB996E   **The** *<parameter>* **table cannot have an incomplete table definition.**

Explanation: In order to enable archiving, the archive and archive-enabled tables must not have an incomplete table definition.

System action: Processing stops.

User response: Assure the tables are defined as complete.

ADB997E   **The** *<parameter>* **table cannot contain a security label column.**

Explanation: In order to enable archiving, neither the archive nor the archive-enabled table can contain a security label column.

System action: Processing stops.

User response: Assure the table does not contain a security label column.

ADB998E   **The** *<parameter>* **table cannot be involved in a clone relationship.**

Explanation: In order to enable archiving, neither the archive nor the archive-enabled table can be involved in a clone relationship.

System action: Processing stops.

User response: Assure the table is not involved in a clone relationship.

ADB999E   **The archive table cannot be** *<parameter>***.**

Explanation: You cannot specify as an archive table a view, a table implicitly created for an XML column, or any of the following:

- Clone table
- Global temporary table
- History table
- MQT
- Auxiliary table
- Existing archive table
- Archive-enabled table
- Catalog table

System action: Processing stops.

User response: Assure the table is not involved in a clone relationship.

ADB0014E   **The input from the PARMS file is not valid. Comments are not allowed in the input file. The invalid input is** `'text_that_is_invalid'`**.**

Explanation: The invalid input that is displayed in the

message contains the text that most likely contains a comment.

System action: Processing stops. Additional errors in the input are not reported.

User response: Check the input file and verify that no comments exist.

ADB0015E   **The input from the PARMS file is not valid. A parameter name might be misspelled. The invalid input is** `'text_that_is_invalid'`**.**

Explanation: The invalid input that is displayed in the message contains the text that likely contains a misspelled parameter name.

System action: Processing stops. Additional errors in the input are not reported.

User response: Check the input file and verify that all the parameter names are spelled correctly.

ADB0016E   **The input from the PARMS file is not valid. The first character of the invalid input is** *first_character* **and the hexadecimal value of this character is** *hexadecimal_value_of_first_character***. If the character is not displayed, check the hexadecimal value. The invalid input is** `'text_that_is_invalid'`**.**

Explanation: A character was detected in a location in the file that is not allowed by the parameter syntax.

System action: Processing stops. Additional errors in the input are not reported.

User response: Verify input and try again.

Related concepts:

"Parameter syntax for Change Management batch interface" on page 602
The following sections describe how the Change Management batch interface parameter syntax works.

ADB0017E   **An error occurred while reading the input parameters from the PARMS file. The invalid input is** `'text_that_is_invalid'`**.**

Explanation: The exact cause of this error is unknown. The most likely cause is unmatched escape characters for a parameter value. A parameter value must be enclosed with the escape character, which is an apostrophe (').

Remember: Two consecutive escape characters must be used to represent one escape character within a parameter value. Here is an example of an invalid and a valid use of escape characters:

- Invalid: job_card_line_1 = '//TEST1234 JOB (INFO),'TEST''

- Valid: job_card_line_1 = '//TEST1234 JOB (INFO),"TEST"'

System action: Processing stops. Additional errors in the input are not reported.

User response: Verify input and try again.

ADB0380E **Module** *module_name* **- Severe error.** *program_name* **is halted.**

Explanation: The specified module has encountered a severe problem and the specified program has halted.

System action: A return code of 12 is set and processing stops.

User response: An internal error has been detected. Contact IBM Software Support.

ADB1003E **An error occurred while processing DBname=** *requested_database*, **TSname=** *requested_table_space*.

Explanation: An unexpected and unknown processing error occurred. The most recent database or table space that was requested is displayed.

System action: Processing stops.

User response: Look for other messages in the job output that might indicate the cause of the error. Contact IBM Software Support if needed.

ADB1026E **The parameter input file is empty.**

Explanation: The parameter input file is generated by the product.

System action: Processing stops.

User response: If the JCL job step that contains the empty parameter file was generated by the product, contact IBM Software Support.

ADB1031E **DDL cannot be generated for DB2 release** *requested_DB2_release*. **Supported releases are** *mininum_supported_DB2_release* **through** *maximum_supported_DB2_release*.

Explanation:

System action: Processing stops.

User response: Specify a supported DB2 release and try again.

ADB1032E **DDL cannot be generated for DB2 release** *local_DB2_system_release*. **Supported releases are** *minimum_supported_DB2_release* **through** *maximum_supported_DB2_release*.

Explanation:

System action: Processing stops.

User response: Ensure that a DB2 connection exists to a supported DB2 release.

ADB1187E **The exclude specification** *exclude_specification_owner* *exclude_specification_name* **does not exist.**

Explanation: A user-specified exclude specification was not found.

System action: Processing stops.

User response: Ensure that the specified owner and name are correct.

ADB1223E *module_name* **Unexpected sqlcode in:** *error_function*

Explanation: The specified module received an unexpected SQL return code from DB2.

System action: Processing stops.

User response: See the details for the SQL code in the DB2 documentation.

ADB1241E **An unexpected error occurred while processing version scope** *version_scope_qualifier*.*version_scope_name*. **Reason code=***reason_code*

Explanation: Reason codes: 1,3 - Report this error to IBM. 2,4 - Look for other error messages to determine the cause.

System action: Processing stops.

User response: Check the reason code and take the indicated action.

ADB1426E **An internal error occurred. Table** *table_creator*.*table_name* **could not be found in an internal data storage.**

Explanation:

System action: Processing stops.

User response: Contact IBM Software Support.

ADB1429W **Clone table** *clone_schema clone_name* **required that base table** *base_table_schema base_table_name* **exist before the clone can be created.**

Explanation: The GEN function created DDL to add a clone, but the base table is not part of the DDL.

System action: None.

User response: No action is necessary if you do not want the base table included in the DDL. Otherwise, include the base table *base_ table_schema base_ table_ name* and run GEN again.

ADB1456e	**The number of plan dependencies has exceeded the product limit of 32K.**

Explanation:

System action: No system action is taken.

User response: A product limit has been reached. The maximum number of plan dependencies for each plan is 32K. Processing stops.

ADB1457e	**The number of package dependencies has exceeded the product limit of 32K.**

Explanation:

System action: No system action is taken.

User response: A product limit has been reached. The maximum number of package dependencies for each package is 32K. Processing stops.

ADB1458e	**The number of packages has exceeded the product limit of 32K.**

Explanation:

System action: No system action is taken.

User response: A product limit has been reached. The maximum number of packages that can be generated is 32K. Processing stops.

ADB1602E	**No SYSVOLUMES record was found in the DB2 catalog for STOGROUP** *stogroup_name***.**

Explanation:

System action: Processing stops.

User response: Contact IBM Software Support.

ADB1607E	**A SYSDATABASE record was not found for table space** *table_space_name***,** **database***database_name***.**

Explanation: The database name recorded in the SYSTABLESPACE record for the specified table space does not have a SYSDATABASE record in the DB2 catalog.

System action: Processing stops.

User response: Contact IBM Software Support.

ADB1610E	**A table space was not found:** *database_name*.*table_space_name*

Explanation: The SYSTABLESPACE record for the specified table space was not found in the DB2 catalog.

System action: Processing stops.

User response: Contact IBM Software Support.

ADB1613E	**The table associated with an index was not found. The index name is** *index_name***. The table name is** *table_name***.**

Explanation: The SYSTABLES record for the table name recorded in a SYSINDEXES record was not found in the DB2 catalog.

System action: Processing stops.

User response: Contact IBM Software Support.

ADB1614E	**The database associated with an index was not found. The index name is** *index_name***. The database name is** *database_name***.**

Explanation: The SYSDATABASE record for the database name recorded in a SYSINDEXES record was not found in the DB2 catalog.

System action: Processing stops.

User response: Contact IBM Software Support.

ADB1627E	*ADB*2**GEN - Location** *location_name* **is not defined on the local DB2 system.**

Explanation:

System action: Processing stops.

User response: Ensure that the DB2 location name is correct.

ADB1628E	*program_name* **- Location** *location_name* **is not a DB2 for z/OS system. Generate DDL will not work for this location.**

Explanation: The specified program is only supported to run on a DB2 for z/OS system.

System action: Processing stops.

User response: Ensure that the specified location is a DB2 for z/OS system.

ADB1636E	**An internal limit has been reached. The catalog row stack is full.**

Explanation:

System action: Processing stops.

User response: Contact IBM Software Support.

ADB1639E	**An internal error has occurred. An unexpected DB2 catalog row type of** *row_type* **was requested.**

Explanation:

System action: Processing stops.

User response: Contact IBM Software Support.

ADB1646E **An error has occurred while generating DDL for an object.**

Explanation:

System action: Processing stops.

User response: Look for error messages prior to this message for additional information.

ADB1650E **An error has occurred while generating the storage group for database** *database_name*.

Explanation:

System action: Processing stops.

User response: Look for error messages prior to this message for additional information.

ADB1651E **An error has occurred while generating the storage group for table space** *database_name*.*table_space_name*.

Explanation:

System action: Processing stops.

User response: Look for error messages prior to this message for additional information.

ADB1652E **An error has occurred while generating the storage group for index** *index_schema*.*index_name*.

Explanation:

System action: Processing stops.

User response: Look for error messages prior to this message for additional information.

ADB1653E **Storage group** *stogroup_name* **was not found in the DB2 catalog.**

Explanation: A storage group that is associated with a table space or index was not found in the DB2 catalog.

System action: Processing stops.

User response: Contact IBM Software Support.

ADB1658W **Index** *index_creator_v index_name_v* **is being generated because the ROWID column** *ROWID_column_name_v* **on table** *table_creator_v table_creator_name_v* **will be converted from GENERATED ALWAYS to GENERATED BY DEFAULT. Converting the ROWID to GENERATED BY DEFAULT is done to allow the ROWID table data to be loaded back into the table using the DB2 LOAD utility.**

System action: None. GEN processing continues.

User response: None.

ADB1660W **The database was skipped because a temporary database is not supported in DB2 V9 or later versions.**

Explanation: A temporary database is being generated for DB2 V9 function mode, but the DB2 V9 function mode does not support temporary databases. The GEN function will not generate DDL for the temporary database.

System action: None. GEN processing continues.

User response: No action is required.

ADB1661W **Table space** *database table_space* **was skipped because it was implicitly created.**

Explanation: The GEN functon does not generate information for an implicit table space for XML columns.

System action: None. GEN processing continues.

User response: No action is required.

ADB1662W **Table** *table_creator table_name* **was skipped because it is an implicit table that was created for XML columns.**

Explanation: GEN does not generate information for an implicit table space that was created for XML columns.

System action: None. GEN processing continues.

User response: No action is required.

ADB1663W **The owner of** *object_type qualified_object_name* **is a role.**

Explanation: If the object owner should be a role when the object is created, a trusted context must be established when creating the object.

System action: None. GEN processing continues.

User response: Establish a trusted context to create the object with a role as the object owner. You can ignore this message if you do not want a role as the object owner.

ADB1664E **An internal error occurred. Diagnostic text=** *diagnostic information for IBM optional object type optional object qualifier. optional object name optional additional diagnostic text optional additional diagnostic text*.

Explanation: This message is issued for several types of internal errors.

Chapter 26. Troubleshooting **837**

System action: Processing stops.

User response: Contact IBM Software Support.

ADB1666W A SYSAUXRELS row was not found for the DB2 auxiliary table *aux_tbcreator.aux_tbname.* **The DB2 table space will not be generated.**

Explanation: If a row is not found in the SYSAUXRELS catalog table, the relationship between the base table and the auxiliary table is unknown and GEN will not generate the table space of the auxiliary table.

System action: GEN processing continues.

User response: No action is required.

ADB1816E A procedure parameter data type of *data_type_id* **is not yet supported.**

Explanation: An unsupported data type was found for a procedure parameter.

System action: Processing stops.

User response: Contact IBM Software Support.

ADB1837E The value for DSSIZE of a table space is not yet supported.

Explanation:

System action: Processing stops.

User response: Contact IBM Software Support.

ADB1841E A function parameter data type of *data_type_id* **is not yet supported.**

Explanation: An unsupported data type was found for a function parameter.

System action: Processing stops.

User response: Contact IBM Software Support.

ADB1847E A parser error has occurred for the following *statement_type.* **GEN cannot complete the request.**

Explanation: The statement could not be parsed by the DB2 Admin parser. Because the GEN request contained DDL changes (such as masking, change owner, change schema, RUN sqlid, and so on), processing stops. The unformatted DDL is generated as an SQL comment.

System action: Processing stops.

User response: Run GEN again with no DDL changes. If the parser error still occurs then contact IBM Software Support. If the parser error does not occur then ensure that the DDL changes are correct.

ADB1871E An internal limit has been reached. The DDL stack is full.

Explanation:

System action: Processing stops.

User response: Contact IBM Software Support.

ADB1873E Processing ended but not all supplied catrows were used.

Explanation:

System action: Processing continues.

User response: Contact IBM Software Support.

ADB1875E An unexpected error return code was received while a mask was being processed.

Explanation: This error can be an internal error or can be caused by an invalid mask being specified.

System action: Processing stops.

User response: If this message was caused by an invalid mask being specified, it will be preceded by additional related messages. Refer to those messages to attempt to correct the problem. If this message is an internal error (that is, is not preceded by additional related messages), contact IBM Software Support.

ADB1877E An error occurred in the DB2 Admin auth-switching module, RC=*return_code*

Explanation:

System action: Processing stops.

User response: If this message is preceded by additional related messages, refer to those messages for more details about this error condition. If this message is not preceded by additional related messages, contact IBM

ADB1907E An invalid TYPE value of *invalid_type_value* **was specified for the***program_name* **program.**

Explanation:

System action: Processing stops.

User response: If the TYPE parameter was built by the product, contact IBM Software Support. Otherwise, ensure the value for TYPE matches a supported value as documented in the DB2 Admin Users Guide.

ADB1915W   The original DDL for the following object will be generated as it is stored in DB2. Verify the DDL.

Explanation: The internal DDL buffer of the DDL statement the GEN program attempted to create exceeded 2 MB. Since the GEN request did not contain any DDL changes (such as masking, change owner, change schema, or Run SQLID) the original DDL that is stored in DB2 is generated.

System action: None.

User response: Verify the DDL is correct.

ADB1916E   The DDL for the following object cannot be created within the 2 MB limit. GEN cannot complete the request.

Explanation:

System action: No system action is taken.

User response: GEN processing stops. The DDL statement the GEN program attempted to create exceeded the output buffer size. The GEN program will not attempt to generate the original DDL stored in DB2. This is most likely because of at least one of the following: - a request was made to change the DDL (i.e. masking, change owner, change schema, RUN sqlid, etc.) - the object was originally created using an ALTER statement - the object has a table parameter GEN cannot complete the request. Try running GEN again with no DDL change requests.

ADB1917W   Unformatted DDL will be generated for the following object because of an unknown formatter error. Verify the DDL.

Explanation: An unknown internal formatter error occurred. Since the GEN request did not contain any DDL changes (such as masking, change owner, change schema, or Run SQLID), the unformatted DDL is generated..

System action: None.

User response: Verify the DDL is correct.

ADB1918E   An unknown formatter error occurred. GEN cannot complete the request for the following *stmt_type*.

Explanation:

System action: No system action is taken.

User response: An unknown internal formatter error occurred. Since the GEN request contained DDL changes (i.e. masking, change owner, change schema, RUN sqlid, etc.), processing stops. GEN cannot complete the request. Try running GEN again with no

DDL changes. The unformatted DDL is generated but as an SQL comment.

ADB1919W   Unformatted DDL will be generated for the following object because the formatted DDL exceeded 2 MB. Verify the DDL.

Explanation: The output formatter buffer size was exceeded. Since the GEN request did not contain any DDL changes (such as masking, change owner, change schema, or Run SQLID), the unformatted DDL is generated.

System action: None.

User response: Verify the DDL is correct.

ADB1920E   The formatted DDL has exceeded 2 MB. GEN cannot complete the request for the following *stmt_type*.

Explanation:

System action: No system action is taken.

User response: The output formatter buffer size was exceeded. Since the GEN request contained DDL changes (i.e. masking, change owner, change schema, RUN sqlid, etc.), processing stops. GEN cannot complete the request. Try running GEN again with no DDL changes. The unformatted DDL is generated but as an SQL comment.

ADB1933E   The DB2 Admin parser could not parse a statement. An SQL comment containing the original DDL will be generated.

Explanation: The DDL statement that the GEN program attempted to create encountered a parser error. GEN cannot complete the request.

System action: Processing stops.

User response: Look for other messages that identify the object being parsed. Try running GEN again with no DDL change requests.

ADB1935E   SQL body not found. GEN cannot complete the request. An SQL comment containing the original DDL will be generated.

Explanation: The DB2 Admin parser could not locate the SQL body in the original DDL text. The GEN program will not attempt to generate the original DDL stored in DB2. This is most likely because one or more of the following: - A request was made to change the DDL, for example, masking, change owner, change schema, and RUN sqlid. - The object was originally created using an ALTER statement. - The object has a table parameter.

ADB1943E • ADB1956E

System action: Processing stops.

User response: Try running GEN again with no DDL change requests. Contact IBM Software Support if needed.

ADB1943E **The "Only" value cannot be specified for both the "Generate catalog stats" and "Include DB2 pending chgs" options.**

Explanation: Choosing "Only" for the specified options is mutually exclusive.

System action: Processing stops.

User response: Specify "Only" for one of the identified options but not both.

ADB1944E **The SYSTABLEPART table contains a record of PARTITION** *part_num* **of** *obj_type obj_qual.obj_name*, **which has an invalid value "**err_value**" for part** *err_seqno* **of column LIMITKEY.**

Explanation: An attempt was made to process the value of a limit key but an unexpected and presumed invalid value was encountered.

System action: Processing stops.

User response: Contact IBM Software Support.

ADB1945W **The INLINE LENGTH** *length* **clause for the column** *column_name* **in table** *table_name* **is not generated because the zparm SPRMRRF is set to disable.**

Explanation: The DB2 zparm SPRMRRF is set to disable. When zparm SPRMRRF is disabled, INLINE LENGTH clauses for columns are not generated.

System action: Processing continues.

User response: Add INLINE LENGTH *length* clauses manually, if needed.

ADB1950E **The "Only" value cannot be specified for both the "Generate index cleanup" and "Include DB2 pending chgs" options.**

Explanation: Choosing "Only" for the specified options is mutually exclusive.

System action: Processing stops.

User response: Specify "Only" for one of the identified options but not both.

ADB1951E **An error occurred when the Gen component called the ADB2ZP program to get the DB2 system parameter (DSNZPARM) values.**

Explanation:

System action: No system action is taken.

User response: See the error that was written in the log file by the ADB2ZP program. Resolve the problem and retry.

ADB1952W **An error occurred when the Gen component called the ADB2ZP program to get the DB2 system parameter (DSNZPARM) values.**

Explanation: However, the DSNZPARM values are not needed because no request was made to remove the default values or generate ADMIN ALTER IMPLICIT statements.

System action: The error is ignored and processing continues.

User response: See the error that was written in the log file by the ADB2ZP program. Resolve the problem and retry.

ADB1953E **ALTER TABLE DROP COLUMN statements were generated for DB2 pending changes. These statments cannot be run on the specified DB2 level. All DDL statements are generated but GEN ends with RC=12.**

Explanation: A DROP COLUMN DB2 pending change exists and a value other than "No" was specified for the "Include DB2 pending changes" option. This results in an ALTER TABLE DROP COLUMN statement being generated that is not supported on the DB2 level specified for the "Target DB2 version" option.

System action: All DDL is generated but GEN ends with RC=12.

User response: To avoid this condition, specify "Target DB2 version" 1115 or higher, or complete or DROP the DB2 pending changes before running GEN.

ADB1956E **An unsupported ARRAYINDEXTYPE value (** *arrayindextypeid_value***) was found in a SYSDATATYPES record.**

Explanation: The value ARRAYINDEXTYPE is not supported.

System action: Processing stops.

User response: Verify that the version of GEN is supported on this version of DB2 and that the value of ARRAYINDEXTYPE is valid.

ADB1957E   **The option GETDB2ZP='N' is specified; therefore GEN cannot call the DB2 stored procedure DSNWZP and get the DB2 system parameter (DSNZPARM) values. The DSNZPARM values are required when GEN generates a version file.**

Explanation: The DB2 system parameter (DSNZPARM) values are needed when writing a version file. The DSNZPARM values are required by downstream functions.

System action: Processing stops.

User response: Specify YES for the option 'Get DB2 ZPARM' in the Change DB2 Admin Defaults panel (ADB2P2).

ADB1958W   **The option GETDB2ZP='N' is specified; therefore GEN cannot call the DB2 stored procedure DSNWZP and get the DB2 system parameter (DSNZPARM) values. The following DSNZPARM values will be used when removing DDL default values and generating ADMIN ALTER IMPLICIT statements: TBSBPOOL=BPP0; TBSBP8K=BP8K0; TBSBP16K=BP16K0; TBSBP32K=BP32K; TBSBPLOB=BP0;TBSBPXML=BP16K0; IDXBPOOL=BP0; WLMENV=''; PADIX=NO; IMPTSCMP=NO; LOB_INLINE_LENGTH=0; IMPTSCMP = NO; MAX_UTIL_PARTS=''; RRF=TRUE.**

Explanation: The DB2 system parameter (DSNZPARM) values are needed when a request is made to remove default values or generate ADMIN ALTER IMPLICIT statements.

System action: Processing continues.

User response: If needed, specify YES for the option 'Get DB2 ZPARM' in the Change DB2 Admin Defaults panel (ADB2P2).

ADB2000I   **Parameter name:** *name*. **Valid values:** *values*

Explanation: The message lists valid values for the specified parameter.

User response: None required.

ADB3000E   **An error occurred while processing the** *object_name* **object in the statement type of** *stmt_type*. **Object already exists.**

Explanation: The syntax of the generated SQL statements is being validated to check consistency. This message is written to the Validate report to indicate an error with the identified SQL statement.

System action: After the Validate report is generated, a return code of 8 is set, and processing stops.

User response: If you are validating a work statement list (WSL), correct the SQL statement in the WSL and re-validate the WSL. If you are using DB2 Object Comparison Tool to compare objects, correct the SQL statement in the DDL source and re-generate the compare job.

ADB3000I   **Default message - At least one message must exist in a message module. This message can be changed.**

Explanation: This is a comment that can be used to explain the message. This comment tag is optional and is not displayed with the message.

System action: This is a comment that can be used to explain the system action. This tag is optional and is not displayed with the message.

User response: This is a comment that can be used to explain the programmer response. This tag is optional and is not displayed with the message.

ADB3001E   **An error occurred while processing the** *object_name* **object in the statement type of** *stmt_type*. **Object does not exist.**

Explanation: The syntax of the generated SQL statements is being validated to check consistency. This message is written to the Validate report to indicate an error with the identified SQL statement.

System action: After the Validate report is generated, a return code of 8 is set, and processing stops.

User response: If you are validating a work statement list (WSL), correct the SQL statement in the WSL and re-validate the WSL. If you are comparing objects with DB2 Object Comparison Tool, correct the SQL statement in the DDL source and re-generate the compare job.

ADB3002E   **An error occurred while processing the** *object_name* **object in the statement type of** *stmt_type*. **The object does not exist.**

Explanation: The syntax of the generated SQL statements is being validated to check consistency. This message is written to the Validate report to indicate an error with the identified SQL statement.

System action: After the Validate report is generated, a return code of 8 is set, and processing stops.

User response: If you are validating a work statement list (WSL), correct the SQL statement in the WSL and re-validate the WSL. If you are comparing objects with DB2 Object Comparison Tool, correct the SQL statement in the DDL source and re-generate the compare job.

ADB3003E **An error occurred while processing the** *object_name* **object in the statement type of** *stmt_type*. **A clustering index already exists on** *object_name2*.

Explanation: The syntax of the generated SQL statements is being validated to check consistency. This message is written to the Validate report to indicate an error with the identified SQL statement.

System action: After the Validate report is generated, a return code of 8 is set, and processing stops

User response: If you are validating a work statement list (WSL), correct the SQL statement in the WSL and re-validate the WSL. If you are comparing objects with DB2 Object Comparison Tool, correct the SQL statement in the DDL source and re-generate the compare job.

ADB3004E **An error occurred while processing object name** *object_name* **in statement type** *statement_type*. **The object was dropped many times.**

Explanation: The syntax of the generated SQL statements is being validated to check consistency. The object was dropped many times.

System action: Processing stops.

User response: If you are validating a work statement list (WSL), correct the SQL statement in the WSL and re-validate the WSL. If you are using DB2 Object Comparison Tool to compare objects, correct the SQL statement in the DDL source and re-generate the compare job.

ADB3004W **An error occurred while processing the** *obj_name* **object in the statement type of** *stmt_typ*. **Multiple attempts were made to drop the object, but the object cannot be dropped.**

Explanation: The syntax of the generated SQL statements is being validated to check consistency. This message is written to the Validate report to indicate an error with the identified SQL statement.

System action: After the Validate report is generated, a return code of 8 is set, and processing continues.

User response: If you are validating a WSL, correct the SQL statement in the work statement list (WSL) and re-validate the WSL. If you are comparing objects with DB2 Object Comparison, correct the SQL statement in the DDL source and re-generate the compare job.

ADB3005E **An error occurred while processing the** *object_name* **object in the statement type** *stmt_typ*. **The object and a foreign key must exist.**

Explanation: The foreign key for the object cannot be found. The syntax of the generated SQL statements is

being validated to check consistency. This message is written to the Validate report to indicate an error with the identified SQL statement.

System action: After the Validate report is generated, a return code of 8 is set, and processing stops.

User response: Ensure that the object and foreign key for the object exists. If you are validating a work statement list (WSL), correct the SQL statement in the WSL and re-validate the WSL. If you are comparing objects with DB2 Object Comparison Tool, correct the SQL statement in the DDL source and re-generate the compare job.

ADB3006E **An error occurred while processing the** *obj_name* **object in the statement type of** *stmt_typ*. **The column** *obj_name2* **does not exist in the table.**

Explanation: The syntax of the generated SQL statements is being validated to check consistency. This message is written to the Validate report to indicate an error with the identified SQL statement.

System action: After the Validate report is generated, a return code of 8 is set, and processing stops.

User response: If you are validating a work statement list (WSL), correct the SQL statement in the WSL and re-validate the WSL. If you are comparing objects with DB2 Object Comparison Tool, correct the SQL statement in the DDL source and re-generate the compare job.

ADB3007E **An error occurred while processing the** *obj_name* **object in the statement type of** *stmt_typ*. **The column** *obj_name2* **is not part of the parent table primary key.**

Explanation: The column that is referenced against the parent table primary key does not exist. The syntax of the generated SQL statements is being validated to check consistency. This message is written to the Validate report to indicate an error with the identified SQL statement.

System action: After the Validate report is generated, a return code of 8 is set, and processing stops.

User response: If you are validating a work statement list (WSL), correct the SQL statement in the WSL and re-validate the WSL. If you are comparing objects with DB2 Object Comparison Tool, correct the SQL statement in the DDL source and re-generate the compare job.

ADB3008E **An error occurred while processing the** *obj_name* **object in the statement type of** *stmt_type*. **The referenced key has been dropped.**

Explanation: The syntax of the generated SQL statements is being validated to check consistency. This message is written to the Validate report to indicate

idate Report to indicate an error with the identified SQL statement.

System action: After the Validate report is generated, a return code of 8 is set, and processing stops.

User response: If you are validating a work statement list (WSL), correct the SQL statement in the WSL and re-validate the WSL. If you are comparing objects withDB2 Object Comparison Tool, correct the SQL statement in the DDL source and re-generate the compare job.

ADB3009E An error occurred while processing the *obj_name* object in the statement type of *stmt_type*. The number of index partitions does not match the number of table space partitions.

Explanation: The syntax of the generated SQL statements is being validated to check consistency. This message is written to the Validate report to indicate an error with the identified SQL statement.

System action: After the Validate report is generated, a return code of 8 is set, and processing stops.

User response: If you are validating a work statement list (WSL), correct the SQL statement in the WSL and re-validate the WSL. If you are comparing objects with DB2 Object Comparison Tool, correct the SQL statement in the DDL source and re-generate the compare job.

ADB3010E An error occurred while processing the *obj_name* object in the statement type of *stmt_typ*. The referenced column *obj_name* does not exist in the parent table.

Explanation: The syntax of the generated SQL statements is being validated to check consistency. This message is written to the Validate report to indicate an error with the identified SQL statement.

System action: After the Validate report is generated, a return code of 8 is set, and processing stops.

User response: If you are validating a work statement list (WSL), correct the SQL statement in the WSL and re-validate the WSL. If you are comparing objects with DB2 Object Comparison Tool, correct the SQL statement in the DDL source and re-generate the compare job.

ADB3011E An error occurred while processing the *obj_name* object in the statement type of *stmt_typ*. The table space is partitioned but a partitioning index has not been found.

Explanation: The syntax of the generated SQL statements is being validated to check consistency. This message is written to the Validate report to indicate an error with the identified SQL statement..

System action: After the Validate report is generated, a return code of 8 is set, and processing stops.

User response: If you are validating a work statement list (WSL), correct the SQL statement in the WSL and re-validate the WSL. If you are comparing objects with DB2 Object Comparison Tool , correct the SQL statement in the DDL source and re-generate the compare job.

ADB3012E An error occurred while processing the *obj_name* object in the statement type of *stmt_typ*. The primary index or the index that is enforcing unique constraint does not have a matching primary or unique key.

Explanation: The syntax of the generated SQL statements is being validated to check consistency. This message is written to the Validate report to indicate an error with the identified SQL statement.

System action: After the Validate report is generated, a return code of 8 is set, and processing stops.

User response: If you are validating a work statement list (WSL), correct the SQL statement in the WSL and re-validate the WSL. If you are comparing objects with DB2 Object Comparison Tool, correct the SQL statement in the DDL source and re-generate the compare job.

ADB3013E An error occurred while processing the *obj_name* object in the statement type of *stmt_typ*. The primary key or unique key does not have a matching primary index or index enforcing unique constraint.

Explanation: The syntax of the generated SQL statements is being validated to check consistency. This message is written to the Validate report to indicate an error with the identified SQL statement.

System action: After the Validate report is generated, a return code of 8 is set, and processing stops.

User response: If you are validating a work statement list (WSL), correct the SQL statement in the WSL and re-validate the WSL. If you are comparing objects with DB2 Object Comparison Tool, correct the SQL statement in the DDL source and re-generate the compare job.

ADB3014E An error occurred while processing the *obj_name* object in the statement type of *stmt_typ*. The column *obj_name2* does not exist in the table or the table does not exist, nor is the column name a known global variable..

Explanation: The syntax of the generated SQL statements is being validated to check consistency. This message is written to the Validate report to indicate an error with the identified SQL statement.

System action: After the Validate report is generated,

a return code of 8 is set, and processing stops.

User response: If you are validating a work statement list (WSL), correct the SQL statement in the WSL and re-validate the WSL. If you are comparing objects with DB2 Object Comparison Tool to compare objects, correct the SQL statement in the DDL source and re-generate the compare job.

ADB3015E **An error occurred while processing the** *obj_name* **object in the statement type of** *stmt_typ***. The object** *obj_name2* **does not exist; it. The object has been renamed.**

Explanation: The syntax of the generated SQL statements is being validated to check consistency. Object name *object_name* in statement type *statement_type* does not exist; it has been renamed.

System action: After the Validate report is generated, a return code of 8 is set, and processing stops.

User response: If you are validating a work statement list (WSL), correct the SQL statement in the WSL and re-validate the WSL. If you are comparing objects with DB2 Object Comparison Tool, correct the SQL statement in the DDL source and re-generate the compare job.

ADB3016E **An error occurred while processing the** *obj_name* **in the statement type of** *stmt_typ***. The object** *obj_name2* **already exists.**

Explanation: The syntax of the generated SQL statements is being validated to check consistency. This message is written to the Validate report to indicate an error with the identified SQL statement.

System action: After the Validate report is generated, a return code of 8 is set, and processing stops.

User response: If you are validating a work statement list (WSL), correct the SQL statement in the WSL and re-validate the WSL. If you are comparing objects with DB2 Object Comparison Tool, correct the SQL statement in the DDL source and re-generate the compare job.

ADB3017E **An error occurred while processing the** *obj_name* **object in the statement type of** *stmt_typ***. The object** *obj_name2* **does not exist.**

Explanation: An attempt was made to drop a clone table, but the specified base table does not have a clone table, or the clone table has been dropped. The syntax of the generated SQL statements is being validated to check consistency. This message is written to the Validate report to indicate an error with the identified SQL statement.

System action: After the Validate report is generated, a return code of 8 is set, and processing stops.

User response: If you are validating a WSL, correct

the SQL statement in the work statement list (WSL) and re-validate the WSL. If you are comparing objects with DB2 Object Comparison Tool, correct the SQL statement in the DDL source and re-generate the compare job.

ADB3020W **The** *obj_name* **object in the statement type of** *stmt_type* **and that are referred in CREATE, ALTER, COMMENT, DROP, EXCHANGE, LABEL, or RENAME statements might not exist during NSP run time.**

Explanation: The syntax of the generated SQL statements is being validated to check consistency. This message is written to the Validate report to indicate an error with the identified SQL statement.

System action: After the Validate report is generated, a return code of 8 is set, and processing stops.

User response: If you are validating a work statement list (WSL), correct the SQL statement in the WSL and re-validate the WSL. If you are comparing objects with DB2 Object Comparison Tool, correct the SQL statement in the DDL source and re-generate the compare job.

ADB3021E **An error occurred while processing the** *obj_name* **object in the statement type of** *stmt_typ***.** *obj_name2* **is not registered in the XML Schema Repository.**

Explanation: The syntax of the generated SQL statements is being validated to check consistency. This message is written to the Validate report to indicate an error with the identified SQL statement.

System action: After the Validate report is generated, a return code of 8 is set, and processing stops.

User response: If you are validating a work statement list (WSL), correct the SQL statement in the WSL and re-validate the WSL. If you are comparing objects with DB2 Object Comparison Tool, correct the SQL statement in the DDL source and re-generate the compare job.

ADB3022E **An error occurred while processing the** *obj_name* **object in the statement type of** *stmt_typ***. The object is a history table and cannot be explicitly dropped.**

Explanation: The syntax of the generated SQL statements is being validated to check consistency. This message is written to the Validate Report to indicate an error with the identified SQL statement.

System action: After the Validate Report is generated, a return code of 8 is set, and processing stops.

User response: If you are validating a work statement list (WSL), correct the SQL statement in the WSL and re-validate the WSL. If you are comparing objects with DB2 Object Comparison Tool, correct the SQL statement in the DDL source and re-generate the compare job.

ADB3023E   An error occurred while processing the *obj_name* **object in the statement type of** *stmt_typ***.**

Explanation: The syntax of the generated SQL statements is being validated to check consistency. This message is written to the Validate Report to indicate an error with the identified SQL statement.

System action: After the Validate Report is generated, a return code of 8 is set, and processing stops.

User response: If you are validating a work statement list (WSL), correct the SQL statement in the WSL and re-validate the WSL. If you are comparing objects with DB2 Object Comparison Tool, correct the SQL statement in the DDL source and re-generate the compare job.

ADB3024E   An error occurred while processing the *obj_name* **object in the statement type of** *stmt_typ***.**

Explanation: The syntax of the generated SQL statements is being validated to check consistency. This message is written to the Validate Report to indicate an error with the identified SQL statement.

System action: After the Validate Report is generated, a return code of 8 is set, and processing stops.

User response: If you are validating a work statement list (WSL), correct the SQL statement in the WSL and re-validate the WSL. If you are comparing objects with DB2 Object Comparison Tool, correct the SQL statement in the DDL source and re-generate the compare job.

ADB3025E   An error occurred while processing the *obj_name* **object in the statement type of** *stmt_typ***. The column** *obj_name2* **already exists in the table.**

Explanation: The syntax of the generated SQL statements is being validated to check consistency. This message is written to the Validate Report to indicate an error with the identified SQL statement.

System action: After the Validate Report is generated, a return code of 8 is set, and processing stops.

User response: If you are validating a work statement list (WSL), correct the SQL statement in the WSL and re-validate the WSL. If you are comparing objects with DB2 Object Comparison Tool, correct the SQL statement in the DDL source and re-generate the compare job.

ADB3026E   An error occurred while processing the *obj_name* **object in the statement type of** *stmt_typ***. The column** *obj_name2* **does not exist in the table or is defined as a NOT NULL column.**

Explanation: The syntax of the generated SQL statements is being validated to check consistency. This message is written to the Validate Report to indicate an

error with the identified SQL statement.

System action: After the Validate Report is generated, a return code of 8 is set, and processing stops.

User response: If you are validating a work statement list (WSL), correct the SQL statement in the WSL and re-validate the WSL. If you are comparing objects with DB2 Object Comparison Tool, correct the SQL statement in the DDL source and re-generate the compare job.

ADB3027E   An error occurred while processing the *obj_name* **object in the statement type of** *stmt_typ***. The EXCLUDE NULL KEYS clause is ignored with UNIQUE indexes.**

Explanation: The syntax of the generated SQL statements is being validated to check consistency. This message is written to the Validate Report to indicate an error with the identified SQL statement.

System action: After the Validate Report is generated, a return code of 8 is set, and processing stops.

User response: If you are validating a work statement list (WSL), correct the SQL statement in the WSL and re-validate the WSL. If you are comparing objects with DB2 Object Comparison Tool, correct the SQL statement in the DDL source and re-generate the compare job.

ADB3028E   An error occurred while processing the *obj_name* **object in the statement type of** *stmt_typ***. The EXCLUDE NULL KEYS clause cannot be specified if a BUSINESS_TIME WITHOUT OVERLAPS index is also specified.**

Explanation: The syntax of the generated SQL statements is being validated to check consistency. This message is written to the Validate Report to indicate an error with the identified SQL statement.

System action: After the Validate Report is generated, a return code of 8 is set, and processing stops.

User response: If you are validating a work statement list (WSL), correct the SQL statement in the WSL and re-validate the WSL. If you are comparing objects with DB2 Object Comparison Tool, correct the SQL statement in the DDL source and re-generate the compare job.

ADB3029E   An error occurred while processing the *obj_name* **object in the statement type of** *stmt_typ***. The EXCLUDE NULL KEYS clause cannot be specified if the index is defined with an XML-index-specification.**

Explanation: The syntax of the generated SQL statements is being validated to check consistency. This message is written to the Validate Report to indicate an error with the identified SQL statement.

System action: After the Validate Report is generated,

a return code of 8 is set, and processing stops.

User response: If you are validating a work statement list (WSL), correct the SQL statement in the WSL and re-validate the WSL. If you are comparing objects with DB2 Object Comparison Tool, correct the SQL statement in the DDL source and re-generate the compare job.

ADB3030E **An error occurred while processing the** *obj_name* **object in the statement type of** *stmt_typ*. **The EXCLUDE NULL KEYS clause cannot be specified if the index is defined with a key-expression.**

Explanation: The syntax of the generated SQL statements is being validated to check consistency. This message is written to the Validate report to indicate an error with the identified SQL statement.

System action: After the Validate report is generated, a return code of 8 is set, and processing stops.

User response: If you are validating a work statement list (WSL), correct the SQL statement in the WSL and re-validate the WSL. If you are comparing objects with DB2 Object Comparison Tool, correct the SQL statement in the DDL source and re-generate the compare job.

ADB3031E **An error occurred while processing the** *obj_name* **object in the statement type of** *stmt_typ*. **The EXCLUDE NULL KEYS clause cannot be specified if the index is defined with an INCLUDE (column name) clause.**

Explanation: The syntax of the generated SQL statements is being validated to check consistency. This message is written to the Validate report to indicate an error with the identified SQL statement.

System action: After the Validate report is generated, a return code of 8 is set, and processing stops.

User response: If you are validating a work statement list (WSL), correct the SQL statement in the WSL and re-validate the WSL. If you are comparing objects with DB2 Object Comparison Tool, correct the SQL statement in the DDL source and re-generate the compare job.

ADB3032E **An error occurred while processing the** *obj_name* **object in the statement type of** *stmt_typ*. **The EXCLUDE NULL KEYS clause cannot be specified if the index is defined as a partitioning index.**

Explanation: The syntax of the generated SQL statements is being validated to check consistency. This message is written to the Validate report to indicate an error with the identified SQL statement.

System action: After the Validate report is generated, a return code of 8 is set, and processing stops.

User response: If you are validating a work statement

list (WSL), correct the SQL statement in the WSL and re-validate the WSL. If you are comparing objects with DB2 Object Comparison Tool, correct the SQL statement in the DDL source and re-generate the compare job.

ADB3033E **An error occurred while processing the** *obj_name* **object in the statement type of** *stmt_typ*. **Add column** *obj_name2*. **The requested operation or usage does not apply to the created global temporary table.**

Explanation: The syntax of the generated SQL statements is being validated to check consistency. This message is written to the Validate report to indicate an error with the identified SQL statement.

System action: After the Validate report is generated, a return code of 8 is set, and processing stops.

User response: If you are validating a work statement list (WSL), correct the SQL statement in the WSL and re-validate the WSL. If you are comparing objects with DB2 Object Comparison Tool, correct the SQL statement in the DDL source and re-generate the compare job.

ADB3034E **An error occurred while processing the** <object_name> <object_type> **in the** <statement_type> **statement. The** <object_type> <object_name> **is already archive enabled or the wrong type of table is specified to be archive enabled.**

Explanation: The SQL statement referred to in this message specifies an archive table name that is already archive enabled or specifies a table cannot be specified as archive enabled. This error message is written to the Validate Report to indicate an error with the identified SQL statement.

User response: Verify that the correct table is specified. Then, if you are validating a WSL, correct the SQL statement in the work statement list (WSL) and re-validate the WSL. If you are using DB2 Object Comparison to compare objects, correct the SQL statement in the DDL source and re-generate the compare job.

ADB3035E **An error occurred while processing the** <object_name> <object_type> **in the** <statement_type> **statement. The** <object_type> <object_name> **is not archive enabled.**

Explanation: The SQL statement referred to in this message specifies an archive table name that is not archive enabled. This message is written to the Validate Report to indicate an error with the identified SQL statement.

User response: Verify that the correct table is specified. Then, if you are validating a WSL, correct the SQL statement in the work statement list (WSL) and

re-validate the WSL. If you are using DB2 Object Comparison to compare objects, correct the SQL statement in the DDL source and re-generate the compare job.

ADB3036E   **An error occurred while processing** *<stmt_type> <obj_type>* **statement:** *<err_msg>*

Explanation: The SQL statement referred to in this message is invalid because of the specified reason. This message is written to the VALOUT data set to indicate an error with the identified SQL statement.

System action: After the Validate Report is generated, a return code of 8 is set, and processing stops.

User response: Correct the SQL statement in the work statement list(WSL) and re-validate the WSL. If you are using DB2 Object Comparison to compare objects, correct the SQL statement in the DDL source and re-generate the compare job.

ADB3037E   **The ADB2IRXCA function** *<stmt_type> <obj_type>* **failed with the following error:** *<err_msg>*

Explanation: The ADB2IRXCA function referred to in this message failed with the specified message.

System action: Processing stops.

ADB3101E   **Unexpected sqlcode in** *error_function*.

System action: No system action is taken.

User response: Fix the problem and try again

ADB3201E   **Applying the DBNAME** *obj_name1* **mask results in the creation of an implicit or system-reserved database,** *obj_name2*.

Explanation: The specified DBNAME mask definition results in the creation of an implicit or system-reserved database, which is not valid because the database is not accepted by DB2 *obj_name1* and *obj_name2*.

System action: A return code of 8 is set and processing stops.

User response: Correct the definition of the DBNAME mask, and resubmit the job.

ADB3202W   **The data set name** *obj_name1* **that is referred to in an UNLOAD statement might not exist after masks are applied.**

System action: Processing continues.

User response: Evaluate the masks that you are using to determine their effect on the specified data set. If the data set does not exist after the masks are applied, correct the problem and resubmit the job.

ADB3301E   **The overwrite value for HASHSPC must be numeric followed by character K, M, or G. Overwrite Value =** *text1*.

Explanation: The use of masking was specified, and the value that is specified for HASHSPC is not valid.

System action: Processing stops.

User response: Correct the definition of the mask. If a specific value is specified for HASHSPC, ensure that the value is an integer value that is followed by the character K, M, or G. If a REXX user exit is specified for HASHSPC, ensure that the REXX user exit is coded so that it returns an integer value followed with the character K, M, or G. After the corrections are made, resubmit the job.

ADB3302E   **The overwrite value for TBINLOBL must be numeric and in a valid range. Overwrite Value =** *text1*.

Explanation: The use of masking was specified, and the value that is specified for TBINLOBL is not valid.

System action: Processing stops.

User response: Correct the definition of the mask. If a specific value is specified for TBINLOBL, ensure that the value is an integer value. If a REXX user exit is specified for TBINLOBL, ensure that the REXX user exit is coded so that it returns an integer value. After the corrections are made, resubmit the job.

ADB3303E   **The overwrite value for DTINLOBL must be numeric and in a valid range. Overwrite Value =** *text1*.

Explanation: The use of masking was specified, and the value that is specified for DTINLOBL is not valid.

System action: Processing stops.

User response: Correct the definition of the mask. If a specific value is specified for DTINLOBL, ensure that the value is an integer value. If a REXX user exit is specified for DTINLOBL, ensure that the REXX user exit is coded so that it returns an integer value. After the corrections are made, resubmit the job.

ADB3304E   **The overwrite value for TBINLOBL exceeded the maximum length of a column. Overwrite Value =** *text1*

Explanation: The use of masking was specified, and the value that is specified for TBINLOBL is not valid.

System action: Processing stops.

User response: Correct the definition of the mask. If a specific value is specified for TBINLOBL, ensure that the overwrite value does not exceed the maximum length of a column. If a REXX user exit is specified for TBINLOBL, ensure that the REXX user exit is coded so

that it returns an overwrite value that will not exceed the maximum length of a column. After the corrections are made, resubmit the job.

ADB3305E   The overwrite value for DTINLOBL exceeded the maximum length of a distinct type.

Explanation: The use of masking was specified, and the value that is specified for DTINLOBL is not valid.

System action: Processing stops.

User response: Correct the definition of the mask. If a specific value is specified for DTINLOBL, ensure that the overwrite value does not exceed the maximum length of a distinct type. If a REXX user exit is specified for DTINLOBL, ensure that the REXX user exit is coded so that it returns an overwrite value that will not exceed the maximum length of a distinct type. After the corrections are made, resubmit the job.

ADB3306E   The overwrite value for the HASHSPC mask is not within the valid range. Overwrite Value = *text1*.

Explanation: The use of masking was specified, and the value that is specified for HASHSPC is not within the valid range.

System action: Processing stops.

User response: If a REXX user exit is specified for the HASHSPC mask, ensure that the REXX user exit is coded so that it returns an overwrite value that is in the valid range. After the corrections are made, resubmit the job.

ADB3307E   The character that is specified in the SINGLECH mask is equivalent to a wildcard (*) character. Single character = *text1*.

Explanation: The use of masking was specified, but the character that is specified for the SINGLECH mask is not valid.

System action: Processing stops.

User response: Correct the definition of the mask. After the corrections are made, resubmit the job.

ADB3308E   The character that is specified in the SINGLECH mask is invalid. Single character = *text1*.

Explanation: The use of masking was specified, but the character that is specified for the SINGLECH mask is not valid.

System action: Processing stops.

User response: Correct the definition of the mask. After the corrections are made, resubmit the job.

ADB3309E   The escape character that is specified in the SINGLECH mask is equivalent to a wildcard (*) character or to the specified single character. Escape character = *text1*.

Explanation: The use of masking was specified, but the escape character that is specified for the SINGLECH mask is not valid.

System action: Processing stops.

User response: Correct the definition of the mask. After the corrections are made, resubmit the job.

ADB3310E   The escape character that is specified in the SINGLECH mask is invalid. Escape character = *text1*.

Explanation: The use of masking was specified, but the escape character that is specified for the SINGLECH mask is not valid.

System action: Processing stops.

User response: Correct the definition of the mask. After the corrections are made, resubmit the job.

ADB3311E   The overwrite value for the AUDIT mask is not valid. Overwrite Value = *text1*.

Explanation: The use of masking was specified, but the value that is specified for the AUDIT mask is not valid.

System action: Processing stops.

User response: If a REXX user exit is specified for the AUDIT mask, ensure that the REXX user exit is coded so that it returns an overwrite value of ALL, CHANGES or NONE.

ADB3312E   The overwrite value for the CLOSE, TSCLOSE or IXCLOSE mask is invalid. Overwrite Value = *text1*.

Explanation: The use of masking was specified, but the value that is specified for CLOSE, TSCLOSE or IXCLOSE is not valid.

System action: Processing stops.

User response: Correct the definition of the mask and resubmit the job. If a REXX user exit is specified for the mask, ensure that the REXX user exit is coded so that it returns an overwrite value of YES or NO.

ADB3313E   The overwrite value for the CCSID mask is invalid. Overwrite Value = *text1*.

Explanation: The use of masking was specified, but the value that is specified for the CCSID mask is not valid.

System action: Processing stops.

User response: Correct the definition of the mask and resubmit the job. If a REXX user exit is specified for the CCSID mask, ensure that the REXX user exit is coded so that it returns an overwrite value of ASCII, EBCDIC or UNICODE.

ADB3314E   The mask value for the SYNSCHEMA mask is too long. Overwrite Value = *text1*.

Explanation: The use of masking was specified, but the value that is specified for the SYNSCHEMA mask is too long. The maximum length is 128 characters.

System action: Processing stops.

User response: Correct the definition of the mask and resubmit the job. If a REXX user exit is specified for the SYNSCHEMA mask, ensure that the REXX user exit is coded so that it returns an overwrite value in the valid range.

ADB3315E   The mask type does not support object-specific masking. Mask type = *text1*.

Explanation: Some mask types are not supported for object-specific masking because they either are too general to determine the objects in question, or they do not refer to objects.

System action: Processing stops.

User response: Correct the definition of the mask. Change the mask to be non-object-specific, or change the mask type to a more specific mask type. For example, use TBNAME instead of NAME if masking a specific table object. After the corrections are made, regenerate, and then resubmit the job.

ADB3316E   The object specification of an object-specific mask does not match the format that is required for the object that is being masked by the mask type. Mask type = *text1*.

Explanation: Mask types require either a single qualifier specification or a qualifier and a name specification depending on the object that is being masked.

System action: Processing stops.

User response: Correct the definition of the mask. Change the object specification to match the required specification. For example, TBNAME:TBSCH1.TBNAME:TBNAME,NEWTB requires both TBSCH1 and TBNAME in the object specification. After the corrections are made, regenerate, and then resubmit the job.

ADB3317W   The external name of a Java program cannot be masked due to the length of the name.

Explanation: Java external names that are greater than 128 characters cannot be masked.

System action: Processing continues.

User response: Change the Java external name manually.

ADB3318W   *text1* could not convert characters from CCSID(*text2*) to CCSID(37).

Explanation: The program could not convert the characters to CCSID(37).

System action: Processing continues.

User response: Use a valid CCSID mask value. See the DB2 for z/OS SQL Reference for valid values.

ADB3319W   The mask value for DSSIZE on the table space *text1* was skipped because the table space is type *text2*.

Explanation: The attribute DSSIZE is only valid in a partitioned table space, partition-by-growth table space, range-partitioned universal table space, and LOB table space.

System action: Processing continues. No system action is taken.

User response: None.

ADB3320W   SEGSIZE was masked from 0 to *text1* for table space *text2*. The value might change the table space type.

Explanation: If the original setting for SEGSIZE mask was 0, then the input mask value might change the table space type. For example, specifying the SEGSIZE mask might convert a partitioned table space to a range-partitioned universal table space (UTS). If a table in a UTS has a partitioned index and the partitioned index needs to be created, DB2 might generate a SQLCODE=-662 error during execution.

System action: Processing continues.

User response: If necessary, specify a valid input mask value, regenerate, and resubmit the job.

ADB3321E   The mask name is too long after applying renames from Name = *<old name>* to Newname = *<new name>*.

Explanation: The use of masking or renames is specified. The value that is specified for masking or renames is too long.

System action: Processing stops.

User response: Correct the name that is defined for the mask or renames, and try again. If a REXX user exit is specified for masks, ensure that the REXX user exit is coded so that a value in the valid range is returned. After the corrections are made, regenerate, and resubmit the job.

ADB3322E	The overwrite value for the TRACKMOD is invalid. Overwrite Value = *text1*

Explanation: The use of masking or renames was specified, but the value that is specified for the TRACKMOD mask is not valid.

System action: Processing stops.

User response: Correct the definition of the TRACKMOD mask. If a REXX user exit is specified for the TRACKMOD mask, ensure that the REXX user exit is coded so that it returns an overwrite value that is either YES or NO. After the corrections are made, regenerate, and resubmit the job.

ADB3323E	The overwrite value for the DCAPTURE (DATA CAPTURE) mask is not valid. Overwrite Value = *text1*

Explanation: The use of masking or renames is specified, but the value that is specified for the DCAPTURE mask is not valid.

System action: Processing stops.

User response: Correct the definition of the DCAPTURE mask. If a REXX user exit is specified for DATA CAPTURE, ensure that the REXX user exit is coded so that it returns an overwrite value of NONE or CHANGES. After the corrections are made, regenerate, and resubmit the job.

ADB3324E	The overwrite value for *text1* FREEPAGE is not correct and must be numeric in the range of 0 - 255. Overwrite Value = *text2*.

Explanation: The use of masking was specified, but the value that is specified for the FREEPAGE attribute overwrites FREEPG or TSFREEPG or IXFREEPG is not valid.

System action: Processing is discontinued with return code 12.

User response: Correct the definition of the FREEPG or TSFREEPG or IXFREEPG overwrites and try again. If a specific value is specified for FREEPG or TSFREEPG or IXFREEPG overwrites, ensure that the value is an integer value in the range of 0 - 255. If a REXX user exit is specified for FREEPG or TSFREEPG or IXFREEPG overwrites, ensure that the REXX user exit is coded so that it returns an integer overwrite value in the range of 0 - 255. After the corrections are made, resubmit the job.

ADB3325E	The overwrite value for *text1* is not correct and must be numeric in the range of 0 - 99.

Explanation: The use of masking was specified, and the value that is specified for PCTFREE attribute overwrites PCTFREE or TSPCTFREE or IXPCTFREE is not valid.

System action: Processing is discontinued with return code 12.

User response: Correct the definition of the PCTFREE or TSPCTFREE or IXPCTFREE overwrites and try again. If a specific value is specified for PCTFREEE or TSPCTFREE or IXPCTFREE overwrites, ensure that the value is an integer value in the range of 0 - 99. If a REXX user exit is specified for PCTFREE or TSPCTFREE or IXPCTFREE overwrites, ensure that the REXX user exit is coded so that it returns an integer overwrite value in the range of 0 - 99. After the corrections are made, resubmit the job.

ADB3326E	The overwrite value for *text1* is not correct and must be numeric in the range of 0-2147483647 or SYSTEM. Overwrite Value = *text2*.

Explanation: The use of masking was specified, and the value that is specified for LOCKMAX is not valid.

System action: Processing is discontinued with return code 12.

User response: Correct the definition of LOCKMAX overwrite and try again. If a specific value is specified for LOCKMAX, ensure that the value is an integer value in the range of 0 - 2147483647 or SYSTEM. If a REXX user exit is specified for LOCKMAX, ensure that the REXX user exit is coded so that it returns an integer overwrite value in the range of 0 - 2147483647 or SYSTEM. After the corrections are made, resubmit the job.

ADB3327E	The overwrite value for *text1* is not correct and should be either YES or NO. Overwrite Value = *text2*

Explanation: The use of masking was specified, and the value that is specified for ERASE attribute overwrites ERASE or TSERASE or IXERASE is not valid.

System action: Processing is discontinued with return code 12.

User response: Correct the definition of the ERASE or TSERASE or IXERASE overwrites and try again. If a specific value is specified for ERASE or TSERASE or IXERASE overwrites, ensure that the overwrite value is YES or NO. If a REXX user exit is specified for ERASE, ensure that the REXX user exit is coded so that it returns an overwrite value that is YES or NO. After the corrections are made, resubmit the job.

| ADB3328E | The specified authorization ID, *authorization_id*, is not valid. |

Explanation: The authorization ID for the *authorization_id* or host variable in the SQL SET CURRENT SQLID statement is not your primary authorization ID or one of the associated secondary authorization IDs.

System action: The SET CURRENT SQLID statement cannot be executed. The current SQL ID is not changed..

User response: Correct the error in the statement or contact the security administrator to have the authorization ID defined for your use.

| ADB3329E | The inmask ends or outmask starts with a comma for field *>masktype<*. |

Explanation: The inmask value ends with a comma for MASK field *>masktype<* or the outmask value starts with a comma for MASK field *>masktype<*.

System action: Processing stops.

User response: Remove the comma.

| ADB3331E | The overwrite value for LOGGED is not correct and should be either YES or NO. |

Explanation: The use of masking was specified, and the value that is specified for LOGGED attribute overwrites is not valid.

System action: Processing is discontinued with return code 12.

User response: Correct the definition of the LOGGED overwrites and try again. If a specific value is specified for LOGGED overwrites, ensure that the overwrite value is YES or NO. If a REXX user exit is specified for LOGGED overwrites, ensure that the REXX user exit is coded so that it returns an overwrite value that is YES or NO. After the corrections are made, resubmit the job.

| ADB3332E | The overwrite value for GBPCACH is not valid. |

Explanation: The use of masking was specified, and the value that is specified for GBPCACH attribute overwrites GBPCACH or TSGBPCACH or IXGBPCACH is not valid.

System action: Processing is discontinued with return code 12.

User response: Correct the definition of the GBPCACH or TSGBPCACH or IXGBPCACH overwrites and try again. If a specific value is specified for GBPCACH or TSGBPCACH or IXGBPCACH overwrites, ensure that the overwrite value is SYSTEM, CHANGED, ALL, or NONE. If a REXX user exit is specified for GBPCACH or TSGBPCACH or

IXGBPCACH overwrites, ensure that the REXX user exit is coded so that it returns an overwrite value that is SYSTEM, CHANGED, ALL, or NONE. After the corrections are made, resubmit the job.

| ADB3333E | The overwrite value for APPEND is not correct and should be either YES or NO. |

Explanation: The use of masking was specified, and the value that is specified for the APPEND attribute overwrite is not valid.

System action: Processing is discontinued with return code 12.

User response: Correct the definition of the APPEND overwrite and try again. If a specific value is specified for the APPEND overwrite, ensure that the overwrite value is YES or NO. If a REXX user exit is specified for APPEND overwrite, ensure that the REXX user exit is coded so that it returns an overwrite value that is YES or NO. After the corrections are made, resubmit the job.

| ADB3334E | The overwrite value for TSPARTS is not correct and must be numeric in the range of 0-4096. |

Explanation: The use of masking was specified, and the value that is specified for TSPARTS is not valid.

System action: Processing is discontinued with return code 12.

User response: Correct the definition of the TSPARTS overwrite and try again. If a specific value is specified for TSPARTS, ensure that the overwrite value is an integer value in the range of 0-4096. If a REXX user exit is specified for TSPARTS, ensure that the REXX user exit is coded so that it returns an integer overwrite value in the range of 0 - 4096. After the corrections are made, resubmit the job.

| ADB3335E | The overwrite value for the LOCKSIZE mask is not valid. |

Explanation: The use of masking was specified, but the value that is specified for the LOCKSIZE mask is not valid.

System action: Processing stops.

User response: Correct the definition of the mask and resubmit the job. If a REXX user exit is specified for the LOCKSIZE mask, ensure that the REXX user exit is coded so that it returns an overwrite value of TABLE, TABLESPACE, LOB, PAGE, ROW, or ANY.

| ADB3336E | The overwrite value for PADDED is not correct and should be either YES or NO. |

Explanation: The use of masking was specified, but the value that is specified for the PADDED attribute overwrite is not valid.

ADB3337E • ADB3343E

System action: Processing is discontinued with return code 12.

User response: Correct the definition of the PADDED overwrite and try again. If a specific value is specified for the PADDED overwrite, ensure that the overwrite value is YES or NO. If a REXX user exit is specified for the PADDED overwrite, ensure that the REXX user exit is coded so that it returns an overwrite value that is YES or NO. After the corrections are made, resubmit the job.

ADB3337E **The overwrite value for MAXROWS is not correct and must be numeric in the range of 0-255.**

Explanation: The use of masking was specified, but the value that is specified for MAXROWS is not valid.

System action: Processing is discontinued with return code 12.

User response: Correct the definition of the MAXROWS overwrite and try again. If a specific value is specified for the MAXROWS overwrite, ensure that the overwrite value is an integer value in the range of 0-255. If a REXX user exit is specified for MAXROWS, ensure that the REXX user exit is coded so that it returns an overwrite value in the range of 0-255. After the corrections are made, resubmit the job.

ADB3338E **The overwrite value for VOLATILE is not correct and should be either YES or NO.**

Explanation: The use of masking was specified, but the value that is specified for VOLATILE is not valid.

System action: Processing is discontinued with return code 12.

User response: Correct the definition of the VOLATILE overwrite and try again. If a specific value is specified for the VOLATILE overwrite, ensure that the overwrite value is YES or NO. If a REXX user exit is specified for VOLATILE, ensure that the REXX user exit is coded so that it returns an overwrite value that is YES or NO. After the corrections are made, resubmit the job.

ADB3339E **The overwrite value for COPY is not correct and should be either YES or NO.**

Explanation: The use of masking was specified, but the value that is specified for COPY is not valid.

System action: Processing is discontinued with return code 12.

User response: Correct the definition of the COPY overwrite and try again. If a specific value is specified for the COPY overwrite, ensure that the overwrite value is YES or NO. If a REXX user exit is specified for COPY, ensure that the REXX user exit is coded so that

it returns an overwrite value that is YES or NO. After the corrections are made, resubmit the job.

ADB3340E **The VER overwrite mask syntax is missing a comma after the word VER.**

Explanation: The use of VER masking was specified, but a comma was not present between the VER string and the overwrite mask.

System action: Processing is discontinued with return code 12.

User response: Correct the syntax of the VER mask and try again. After the corrections are made, resubmit the job.

ADB3341E **The VER overwrite mask syntax has an invalid operand. Operand *text1* mask type was *text2*.**

Explanation: The use of VER masking was specified, but an invalid operand was given. The valid operands are EQ, NE, GT, LT, RANGE and LIST.

System action: Processing is discontinued with return code 12.

User response: Correct the syntax of the VER mask and try again. After the corrections are made, resubmit the job.

ADB3342E **The VER overwrite mask syntax is missing an equal sign after the word RC.**

Explanation: The use of VER masking was specified, but an equal sign was not present after the RC string.

System action: Processing is discontinued with return code 12.

User response: Correct the syntax of the VER mask and try again. After the corrections are made, resubmit the job.

ADB3343E **The VER overwrite mask syntax is missing the word RC.**

Explanation: The use of VER masking was specified, but the RC string was not present.

System action: Processing is discontinued with return code 12.

User response: Correct the syntax of the VER mask and try again. After the corrections are made, resubmit the job.

ADB3344E   The VER overwrite mask syntax is for range operation, but two values are not present after the RANGE string.

Explanation: The use of VER masking was specified, but two values are missing after the RANGE string.

System action: Processing is discontinued with return code 12.

User response: Correct the syntax of the VER mask and try again. After the corrections are made, resubmit the job.

ADB3345E   The VER overwrite mask syntax is for range operation, but more than two values are present after the RANGE string.

Explanation: The use of VER masking was specified, but there are too many values after the RANGE string.

System action: Processing is discontinued with return code 12.

User response: Correct the syntax of the VER mask and try again. After the corrections are made, resubmit the job.

ADB3346E   RC = *return_code*. Error processing *mask_name*. Value is *attr_value*. Verification failed for operation *verify_oper*.

Explanation: The attribute value does not conform to the verification mask.

System action: Processing is discontinued with return code 8 or 12.

User response: Correct the attribute value and try again. After the corrections are made, resubmit the job.

ADB3348E   The VER overwrite mask syntax has an invalid operand for mask *maskname*. The operand is *veroper*.

Explanation: The use of VER masking was specified, but an invalid operand was given. Some operands are not allowed for some masks.

System action: Processing is discontinued with return code 12.

User response: Correct the syntax of the VER mask and try again. After the corrections are made, resubmit the job.

ADB5000E   An invalid value specified for parameter *insert1*.

Explanation: An invalid value was specified for the parameter.

System action: Processing stops.

User response: Specify a valid value for the parameter.

ADB5001E   The `PLAN=` parameter was not found.

Explanation: The `ADBOPT` parameter of `PLAN=` is required for ADBTEPA.

System action: Processing stops.

User response: Provide the `PLAN=` parameter in the ADBOPT DD card.

ADB5002E   The ADBTEPA invocation was not from an APF-authorized environment.

System action: Processing stops.

User response: Use APF to authorize all data sets in the STEPLIB.

ADB5003E   A failure occurred attempting command *RexxCmd1*.

Explanation: The provided command failed for an undetermined reason.

System action: Processing stops.

User response: If possible, resolve the problem and run the *RexxCmd1* command again.

ADB5005E   A DB2 pending change will be lost by dropping the object.

Explanation: The input statement was a DROP, the object involved had a DB2 pending change which would be lost, and the PENDINGCHANGESCHECK='YES' parameter was specified.

System action: Processing stops.

User response: None.

ADB5007E   An invalid input parameter *InvalidParm* was encountered.

Explanation: The parameter is unrecognized.

System action: Processing stops.

User response: Remove the unrecognized parameter.

ADB5008E   Either the `SSID=` or `PLAN=` parameter was not provided.

System action: Processing stops.

User response:

1. Provide the `PLAN=`*plan* setting in the ADBOPT DD card.
2. Provide the `SSID()` parameter as a parameter to the program.

ADB5009E   **A non-zero SQL code was issued.**

System action: Processing stops.

User response: Investigate the specific SQL code and take remedial action.

ADB5010W   **The DB2 Version could not be determined.**

Explanation: This message indicates a positive, non-zero return code from the SQL CONNECT statement.

System action: The product assumes a DB2 Version 6 level, and processing continues.

User response: None.

ADB5011W   **ADBCHKPT update failed for WORKLIST(*Wklist*) during RESTART(NO) processing.**

System action: Processing continues.

User response: None.

ADB5017E   **The ADBCHKPT checkpoint table does not exist.**

System action: Processing stops.

User response: Check the package qualifier of ADBTEP2.

ADB5021W   **The preceding query was cancelled by RLF after successful retrieval of *RecCnt* rows.**

System action: The cursor is closed and processing continues.

User response: None.

ADB5025E   **A "Not Found" condition was encountered during an open.**

System action: Processing continues.

User response: None.

ADB5028E   **An authorization error occurred during -START.**

System action: Processing stops.

User response: Grant the job submitter ID the necessary authority and restart the batch statement list.

ADB5029E   **An error occurred during -START.**

Explanation: An unrecognized error occurred while attempting the **-START** command.

System action: Processing stops.

User response: Examine the output and take remedial action.

ADB5031W   **No statements were found that can be run.**

System action: Processing continues.

User response: None.

ADB5034E   **Delete failed for ADBCHKPT control record for WORKLIST(*WorkList*).**

System action: Processing stops.

User response: Resubmit the job to complete processing.

ADB5035E   **Invalid input parm term character.**

System action: Processing stops.

User response: Specify a valid term character.

ADB5036E   **A trailing parenthesis has been omitted or no value was provided.**

System action: Processing stops.

User response: Specify a trailing parenthesis or provide a value.

ADB5037E   **An error in the MAXE input parameter parenthesis occurred.**

System action: Processing stops.

User response: Specify a trailing parenthesis or provide a value.

ADB5043E   **Restart processing was halted due to a command mismatch.**

Explanation: The command from the last run does not match the command from the restarted run.

System action: Processing stops.

User response: Verify that the statement being restarted has not been changed. Alternatively, you can start the job run with the parameter RESTART(FORCE). ADBTEP2 will skip the changed command and continue the run.

**ADB5051E   An error occurred in the `CHANGEID()`
input parameter**

System action: Processing stops.

User response: Provide the correct `CHANGEID()`
parameter and value.

**ADB5052E   The CONNECT statement contained
syntax errors.**

System action: Processing continues.

User response: Specify a valid CONNECT statement.

**ADB5054E   The SET CONNECTION statement
contained syntax errors.**

System action: Processing continues.

User response: Specify a valid SET CONNECTION
statement.

**ADB5056E   There is an error in the `CHANGEID()` input
parameter value.**

System action: Processing stops.

User response: Provide the correct `CHANGEID()`
parameter and value.

**ADB5057E   The SET QUERYNO statement
contained syntax errors.**

System action: Processing continues.

User response: Specify a valid SET QUERYNO
statement.

**ADB5058E   The RELEASE statement contained
syntax errors.**

System action: Processing continues.

User response: Specify a valid RELEASE statement.

**ADB5063E   The ADBCHKPT control record for
WORKLIST(*WorkList*) is missing.**

System action: Processing stops.

User response: Provide the WORKLIST(*WorkList*)
parameter and value.

**ADB5064E   There is an SQL buffer overflow. The
maximum size is *Maxsize*.**

System action: Processing stops.

User response: Specify a larger region size.

**ADB5067E   The command `Command` is not supported,
or execs are not in SYSEXEC/SYSPROC.**

System action: Processing stops.

User response: Provide a SYSEXEC DD card.

ADB5071E   The ADBPART table does not exist.

System action: Processing stops.

User response: Check the qualifier of package
ADBTEP2.

ADB5073W   Keys do not match for part *PartNo* .

Explanation: Limitkeys do not match between unload
and load. Processing of data might proceed serially.

System action: Processing continues.

User response: None.

ADB5074W   Unloads will be performed using DB2.

Explanation: When a condition is encountered which
requires a DB2 unload, the unload will be performed
by DB2, not by HPU.

System action: Processing continues.

User response: None.

**ADB5080E   A restart with a different unload
method is not allowed.**

Explanation: It is not permitted to change the
UNLOAD method on restart.

System action: Processing continues.

User response: Either resubmit the restart with DB2
(parm `UNLOAD(HPU)` or start the run from the beginning
`RESTART(NO)`.

**ADB5081E   A restart with a different unload
method is not allowed.**

Explanation: It is not permitted to change the
UNLOAD method on restart.

System action: Processing continues.

User response: Either resubmit the restart with DB2
(parm `UNLOAD(DB2)` or start the run from the beginning
`RESTART(NO)`.

**ADB5094E   The held DSN commands have been
queued on SYSIN and will be retried.**

System action: Processing continues.

User response: None.

ADB5100E   **No restart was requested and no checkpoint was found. This was an abnormal run, and cannot be restarted.**

System action: Processing stops..

User response: None.

ADB5105E   **The command `Command` is not supported or the execs are not in SYSEXEC/SYSPROC.**

System action: Processing stops.

User response: Provide a SYSEXEC DD containing the product execs.

ADB5106I   **The following error is tolerated. The value of the parameter MAXERRORS determines the number of errors that are tolerated.**

Explanation: An error occurred but processing continues because the MAXERROR parameter is specified with a value of -1 or a value between 1 and 99.

System action: Processing continues.

User response: If you do not want error tolerance, set the MAXERRORS parameter to 0. Specify a value of -1 to indicate that the program should tolerate an unlimited number of errors for DSN commands. Specify a value between 1 and 99 to indicate the number of errors that the program should tolerate.

ADB5254I   **The SSID parameter that is passed to the program can not be validated. The information that is used for SSID validation can not be obtained because an IFI return code <rc> and a reason code <rc> occurred during the execution of the -DIS GROUP DETAIL DB2 command.**

Explanation: The -DIS GROUP DETAIL command fails, therefore no information can be used to validate the SSID parameter.

System action: Processing continues.

User response: Optionally, if the job fails, verify that the value in the SSID parameter is correct.

ADB5255I   **The SSID parameter that is passed to the program can not be validated. The information that is used for SSID validation is not complete in the output of -DIS GROUP DETAIL. More information, than can be displayed, exists.**

Explanation: The maximum number of subgroup

attachment groups is displayed in the output from executing the -DIS GROUP DETAIL DB2 command. More information exists but cannot be displayed. The SSID is passed to the program but is not validated.

System action: Processing continues.

User response: Optionally, if the job fails, verify that the value in the SSID parameter is correct.

ADB5256I   **The SSID parameter that is passed to the program is not recognized as a DB2 subsystem name in a non-data sharing environment, or as a DB2 subsystem name for a member, group, or subgroup in a data sharing environment.**

Explanation: The SSID parameter that is passed to the program does not match one of DB2 subsystem names, group attachment name or subgroup attachment names in the output from executing the -DIS GROUP DETAIL DB2 command. The SSID problem might cause the job to fail.

System action: Processing continues.

User response: Optionally, if the job fails, verify that the value in the SSID parameter is correct.

ADB5257I   **Table(s) have been successfully recreated on the accelerator.**

Explanation: The specified schema and table name have been successfully loaded.

System action: Processing continues.

User response: None.

ADB5258E   **Table(s) cannot be recreated on the accelerator. An error occurred during the call to the accelerator stored procedure *procedure_name*.**

Explanation: Unexpected error in accelerated table recreate processing.

System action: Processing stops.

User response: Investigate the failure reason and take remedial action.

ADB5263E   **The program *program_name* invocation was not from an authorized program facility (APF) library.**

Explanation: Only load modules from an APF task can invoke the APF protected supervisor calls (SVCs).

System action: Processing stops.

User response: Ensure that the program is APF authorized by making changes to meet the following conditions:

| • The steplib data set name matches the data set name in the APF list.
| • Each data set in the concatenation is APF authorized.
| • The APF list specifies the correct volid.
| • When SMS is specified as the volid in the APF list, the volume shown in the LISTC output is SMS managed.
| • The required module names are listed in the output from PARMLIB.

ADB5264I   **ADB5264I Reason: adb5258e_failed_reason**

Explanation: Shows details about the cause of the error which caused message ADB5258E to be issued.

System action: Processing stops.

User response: Resolve the problem and re-run the job.

ADB5266E   **Table(s) cannot be loaded in the accelerator.**

Explanation: Unexpected error in the accelerated table load processing.

System action: Processing stops.

User response: Investigate the failure reason and take remedial action.

ADB5267E   **Table(s) cannot be deleted from the accelerator.**

Explanation: Unexpected error in the accelerated table delete processing.

System action: Processing stops.

User response: Investigate the failure reason and take remedial action.

ADB5269E   **Table(s) cannot be added to the accelerator.**

Explanation: Unexpected error in the accelerated table add processing.

System action: Processing stops.

User response: Investigate the failure reason and take remedial action.

ADB5271E   **The accelerated tables cannot be table-set in the accelerator.**

Explanation: Unexpected error in the accelerated table enable/disable processing.

System action: Processing stops.

User response: Investigate the failure reason and take remedial action.

ADB5274E   **The table** *table-name* **cannot be archived in the accelerator.**

Explanation: Unexpected error in the accelerated table archive processing.

System action: Processing stops.

User response: Investigate the failure reason and take remedial action.

ADB5299E   **An error occurred while processing the ADMIN UNLOAD statement for the image copy process.**

Explanation: The image copy cannot be processed because the ADMIN UNLOAD failed. The possible cause of failure is indicated by the reason code. See the following list for an explanation of the reason code:

9995   The image copy database or table space was not found.

9996   The image copy destination was not found.

9997   The image copy date or time is in the wrong format.

9999   The ADMIN UNLOAD statement is incomplete or contains a syntax error.

System action: Processing stops.

User response: Correct the ADMIN UNLOAD statement according to the reason code and rerun the job.

ADB5300E   **The pending changes action parameter (PCACT) is set to CANCEL. This prevents the change from being recovered when there are pending changes that modify the same or related objects. To recover this change and set the pending changes to DEFINED status, set the PCACT parameter to SUPERSEDE and re-submit the job (e.g. PCACT(SUPERSEDE)).**

Explanation: The change cannot be recovered when there are pending changes that modify the same or related objects. The pending changes action parameter (PCACT) is set to CANCEL.

System action: Processing stops.

User response: Complete the pending changes first, or change the PCACT parameter to SUPERSEDE. Then, resubmit the job.

ADB5411I   **The RESTART AT statement is ignored and restart will occur at the restart point that is specified in the RESTART() parameter because the restart point name <string_name> in this statement was used previously.**

ADB5412E • ADB6042E

Explanation: If the restart point name in RESTART AT statement was used previously and a different restart point name is specified in the RESTART() parameter, the RESTART() restart point overrides the RESTART AT restart point.

System action: Processing continues.

User response: None.

ADB5412E **The restart point name <string_name> in the RESTART AT statement was used previously.**

Explanation: If the restart point name in the RESTART AT statement was used previously, the program will stop to prevent an unwanted second restart.

System action: Processing stops

User response: Specify parameter RESTART() with the used restart point name to confirm reusing the same restart point, or enter a blank value in the restart_label column of the checkpoint record.

ADB5413E **The restart point name <string_name> in the RESTART AT statement is different from the restart point name <string> in the RESTART() parameter.**

Explanation: The program will stop if the restart point names in the RESTART AT statement and the RESTART() parameter do not match.

System action: Processing stops.

User response: Remove either the RESTART AT statement or remove the RESTART() parameter.

ADB5501E **The DDL file validation date has expired. Create timestamp:** *timestamp***. Validation date:** *date***.**

Explanation: The statements that you can run with the auth-switch ID depend on your authority as defined in the RACF profile that protects the resource. If you have READ authority, the DDL must be run within 8 days of being created.

System action: Processing stops.

User response: Regenerate the DDL file and try again.

ADB5507E **Use of WSL auth-switching was rejected. The submitter does not have ALTER authority to use the RACF profile of <ID>.**

Explanation: Use of WSL auth-switching requires the submitter to have ALTER authority to use the RACF profile.

System action: Processing stops.

User response: Verify the RACF facility setting of

ADBAUTHS and ensure that the submitter has ALTER authority to use the auth-swith ID's profile.

ADB6001W **There is invalid text in file ALTPARM.**

System action: None.

User response: Correct the input parameter in ALTPARM and try again.

ADB6002E **The DD statement** *DDstatement* **is missing or is incorrect.**

System action: Processing stops.

User response: Supply the missing DD statement, and try again.

ADB6003E **Program ADBALT detected an ONCODE condition.**

System action: A return code of 12 is set, and processing stops.

User response: Report this internal error to IBM Software Support.

ADB6025E **Program ADBALT detected an ONCODE condition.**

System action: A return code of 12 is set, and processing stops.

User response: Report this internal error to IBM Software Support.

ADB6026E **Open input error:** *text1***.**

System action: Processing stops.

User response: Correct the open input error for CHGIN and resubmit the job.

ADB6027E **Close input error:** *text1***.**

System action: Processing stops.

User response: Correct the close input error for CHGIN and resubmit the job.

ADB6041W **There is invalid text in file** *CPPARM***.**

System action: None.

User response: Correct the input parameter in CPPARM and try again.

ADB6042E **For one-to-many copy privileges, the specified version scope** *version scope qualifier version scope name* **definition might be empty or the** *NAMES* **does not have any requests to generate GRANT DDLs for the copy privileges command.**

System action: None.

User response: The objects lists generated for processing copy privileges might be empty or the specified version scope definition for one-to-many might be empty or incorrect.

ADB6043E **The source object type specified to copy privileges is invalid.**

System action: Processing stops.

User response: Correct the source type and the try again.

ADB6044E **There are empty input parameters in file** *CPPARM***.**

System action: Processing stops.

User response: Specify input parameters in file *CPPARM* to complete the copy privilege run.

ADB6045E **The catalog row stack is full and the run will terminate.**

System action: Processing stops.

User response: The copy privileges command for one-to-many can accommodate a maximum of 12500 GRANTS for source objects. Contact IBM Software Support.

ADB6046W **For one-to-many copy privileges, the specified quick scope or version scope** *<version scope qualifier>. <version scope name>* **does not have objects that match the specified FROM type** *<FROM object type>***. An empty definition will result in no generated GRANT DDLs for the copy privileges command.**

System action: Processing continues.

User response: The specified TO version scope or quick scope could not find the objects that match the specified FROM type. This results in no GRANTs generated and can lead to an empty file.

ADB6300E **Processing error. The program will now terminate.**

Explanation: An error occurred in processing.

System action: Processing stops.

ADB6310I **No LOAD utility options specified.**

Explanation: LOAD utility options missing.

System action: Processing stops.

User response: Supply the LOAD utility option, and try again.

ADB6311E **The null indicator is set to** *value* **in the HPU configuration, which does not match the default value. Only the default setting is allowed when data conversion is involved.**

Explanation: HPU is used as the unload method, and the HPU PARMLIB parameter VUU014/ULNULL is set to a value that does not match the default value, FF00.

System action: Processing stops.

User response: Change the configuration of HPU to use the default null indicator and rerun the job.

ADB7001W **The REPLACE keyword in the LOAD control statement for table** *table_name* **is converted to RESUME YES. Reason:** *reason_code***.**

Explanation: DB2 restrictions on LOAD REPLACE require a change to the LOAD control statement. The reason code indicates the source of the error:

01 The table to be loaded is a system-period temporal table with data versioning define.

02 The table to be loaded is an archive-enabled table.

03 The table is under a multi-table table space and not all the tables under the table space are migrated.

System action: Processing continues.

User response: Review the LOAD control statement for the specified table, particularly the REPLACE keyword. Correct the statement, if necessary, and try again.

ADB7002W **The LOAD REPLACE option is applied to the multi-table table space** *ts_name***. Any additional tables in the target table space are left empty after migration.**

Explanation: The LOAD REPLACE option is applied to the table space as specified in the LOAD Utility options because all the tables under the table space are selected for migration on the source system. Any additional tables in the table space on the target system are left empty after migration because the LOAD REPLACE option is used.

System action: Processing continues.

User response: Confirm that it is appropriate to use the LOAD REPLACE option before submitting the target jobs.

ADB7100E   SQL statement too long - internal error

System action: Processing stops.

User response: Fix the problem and try again.

ADB7102E   The table *table_name* **contains too many columns.**

Explanation: You can assign up to 750 columns for a non-dependent table. Dependent tables can have up to 749 columns.

System action: Processing stops.

User response: Limit the number of columns to allowed values and try the operation again.

ADB7103E   If ignore partitioning is specified, Object Compare will take partition information from the target. Partitioning is not allowed on partition-by-growth tablespace.

System action: Processing continues.

User response: Fix the problem and try again.

ADB7104W   The *file_name* **data definition is not defined, which can lead to errors due to insufficient sort work file size.**

Explanation: The *file_name* data definition (DD) file, which is an alternate location to store the record count, is not defined. Sometimes use of the *<version file name>* DD file can cause the sort process to underestimate the number of records in the file. Errors can occur due to insufficient sort work file sizes. Take action if the *file_name* is a version file created by GEN or DTC. If the version file is created from change management, you can ignore this information.

System action: Processing stops.

User response: The *file_name* DD is not defined, generate the job again. If the problem persists, make sure that the skeletons are current.

ADB7105E   Substring outside string - internal error.

System action: Processing continues.

User response: Fix the problem and try again.

ADB7106W   *<insert1>* source authorizations for *<insert2> <insert3>* will not be copied to the target because the grantor and grantee are the same. The problem is likely caused by masking.

System action: Processing continues.

User response: Fix the problem and try again.

ADB7107I   *<insert1>* is an index on auxiliary table. It will be kept because the base table *<insert2>* is kept.

System action: Processing continues.

User response: Fix the problem and try again.

ADB7108E   The explicit LOB table space *<insert1>* is still associated with auxiliary table *<insert2>* and therefore cannot be dropped.

System action: Processing continues.

User response: Fix the problem and try again.

ADB7109I   The explicit LOB table space *<insert1>* is supposed to be dropped but is kept because the base table *<insert2>* is kept.

System action: Processing continues.

User response: Fix the problem and try again.

ADB7110I   Change to clustering attribute is ignored because <insert1> can not be clustered.

System action: Processing continues.

User response: Fix the problem and try again.

ADB7112I   *<insert1>* is table partitioned, *<insert2>* is *<insert3>*.

Explanation: The table is being changed either from partitioned to non-partitioned table or vice versa.

System action: Processing continues.

User response: No action is required.

ADB7113I   *<insert1>* change is ignored because the ignore option *<insert2>* is specified.

Explanation: The change is part of the ignore fields specification which is part of this compare run.

System action: Processing continues.

ADB7114W   Base table space has been changed from partitioned to PBG .<insert1> table space will be recreated with DB2 default values.

Explanation: Changing type of the table space to PBG causes the table space and table to be dropped and recreated. All implicit LOB and XML table spaces will be dropped and recreated by DB2 with attributes having default values.

System action: Processing continues.

User response: No action required.

ADB7115E   Invalid parent key of table.

Explanation: The referenced parent key has not been defined as a primary key or a unique key.

System action: Processing stops.

User response: Ensure that the parent key is defined as a primary key or a unique key.

ADB7116E   No match to the child column was found in the corresponding parent table.

Explanation: The referenced parent key does not have the same number of columns as the child key.

System action: Processing stops.

User response: Ensure that each child column as a corresponding parent column

---

ADB7117W   No index was created for the foreign key column.

Explanation: If the foreign key column is not indexed, the performance of DELETE on the parent table may be affected.

System action: Processing continues.

User response: For optimum performance, create an index for the foreign key column.

ADB7118W   The *<insert1>* table *<insert2>* is not in the current scope of analysis. The correctness of foreign key cannot be determined.

System action: Processing continues.

User response: Verify that the parent table is in the catalog.

ADB7120I   The change has been ignored.

Explanation: The change is part of the ignore changes specification which is part of this compare run.

System action: Processing continues.

User response: No action is required.

ADB7121I   The ARRAYINDEXTYPEID of the source and target are different. *var_name* cannot be ignored.

Explanation: The index type of an associative array must be VARCHAR or INTEGER. If the source and target have different index types, then the index length and index subtype cannot be ignored.

System action: Processing continues.

User response: No action is required.

ADB7122I   The change has been ignored because it is related to ignored *<insert1>* change.

Explanation: The change is related to other changes which are part of the ignore specification included in this compare run.

System action: Processing continues.

User response: No action is required.

ADB7122I   The change has been ignored because it is related to ignored *<insert1>* change.

Explanation: The change is related to other changes which are part of the ignore specification included in this compare run.

System action: Processing continues.

User response: No action is required.

ADB7123E   The logging attribute of the LOB table space *lob_tsname* that is associated with the base table *tbname* can not be changed to LOGGED because the logging attribute of the base table space *base_tsname* is NOT LOGGED.

Explanation: If the logging attribute of the base table space is NOT LOGGED, the logging attribute of the LOB table space associated with the base table can not be LOGGED.

System action: Processing continues.

User response: Change the logging attribute of the base table space or the LOB table space and try again.

ADB7124I   The logging attribute of the LOB table space *lob_tsname* that is associated with the base table *tbname* is changed to LOGGED. Information on the logging attribute of the base table space is not available.

Explanation: An inconsistency exists if the logging attribute of the base table space is NOT LOGGED and the logging attribute of the LOB table space that is associated with the base table is LOGGED.

System action: Processing continues.

User response: If needed, change the logging attribute of the base table space or the LOB table space and try again.

ADB7125W   Active versioning is in effect with this base table. In order to add a clone table the base table must be dropped and recreated.

Explanation: The ALTER statement cannot be used to define a clone table because the base table is in a state

of active versioning. Dropping and recreating the table will reset the active versioning and therefore allow you to add the clone table.

System action: Processing continues.

User response: None.

ADB7126I **The active versioning that is in effect with this base table is caused by altering the table during this compare run. In order to add a clone table the base table must be dropped and recreated.**

Explanation: The ALTER statement cannot be used to define a clone table because the base table is in a state of active versioning. Dropping and recreating the table will reset the active versioning and therefore allow you to add the clone table.

System action: Processing continues.

User response: If needed, change the logging attribute of the base table space or the LOB table space and try again.

ADB7127W **The information about base table active versioning is not available. Check whether the base table is in a state of active versioning before applying changes.**

Explanation: The ALTER statement cannot be used to define a clone table if the base table is in a state of active versioning. Because the target is a data set, the information about the active versioning is not available. The clone table will be added via an ALTER TABLE statement without dropping the base table.

System action: Processing continues.

User response: Check whether the base table has an active versioning before applying changes. If active versioning is in effect, reset the versioning by executing REORG followed by MODIFY RECOVERY. Then, run compare. Alternatively, you can run compare using the catalog as the target.

ADB7130W **Clone table** *<insert1>. <insert2>* **is specified in the exclude specification. It will be** *<insert3>***.**

System action: Processing continues.

User response: Fix the problem and try again.

ADB7131W **Clone table** *<insert1>. <insert2>* **is specified in exclude specification. It will not be** *<insert3>***.**

System action: Processing continues.

User response: Fix the problem and try again.

ADB7132W **<Insert1> <insert2>. <insert3> is specified in the <insert4> exclude specification. This object is excluded.**

System action: Processing continues.

ADB7133W **<Insert1> <insert2>. <insert3> is excluded.**

System action: Processing continues.

ADB7134W **History table <insert1>. <insert2> is specified in exclude specification.**

System action: Processing continues.

ADB7135W **Temporal table <insert1>. <insert2> and history table are both excluded.**

System action: Processing continues.

ADB7136E **<insert1>. <insert2> is an excluded object and needs to be implicitly dropped. To allow implicit drop of an excluded object, specify NO. Object Compare is terminated.**

System action: Processing continues.

User response: Fix the problem and try again.

ADB7137I **Exclude specification <insert1>. <insert2> can not be found.**

System action: Processing continues.

User response: Fix the problem and try again.

ADB7138I **<insert1> <insert2>. <insert3> is an excluded object. It will not be dropped.**

System action: Processing continues.

User response: Fix the problem and try again.

ADB7139E **The CREATE INDEX statement may lead to error SQLCODEN662/SQLCODE -662 because the table** *table_name* **on which the index is being created is in the table space** *tablespace_name***. The table space is defined as** *tablespace_type***.**

Explanation: The partitioned index cannot be created on the specified table space, or the table space cannot be index-controlled.

System action: Processing continues.

User response: Verify that you are using the correct table space type for creating a partitioned index. Any changes to the table space type may be due to one of the following conditions:

1. Original definition of the table space was incorrect.

2. Changes to table space attributes SEGSIZE, MAXPARTITIONS, or NUMPARTS were specified.

3. Mask or ignore was specified on table space attributes SEGSIZE, MAXPARTITIONS, or NUMPARTS.

4. Generic ignore PARTITIONING field was specified.

| **ADB7140E** | *<insert1> <insert2>* **is specified in** *<insert3>* **exclude specification. This object is excluded.** |

System action: Processing continues.

User response: Fix the problem and try again.

| **ADB7141I** | *<insert1> <insert2>* **is an excluded object. It will not be dropped.** |

System action: Processing continues.

User response: Fix the problem and try again.

| **ADB7142I** | **Implicit XML Tablespace for target table** *<insert1>. <insert2>* **is excluded because its target table is excluded.** |

System action: Processing continues.

User response: Fix the problem and try again.

| **ADB7143I** | **Implicit index** *<insert1>* **for target table** *<insert2>. <insert3>* **is excluded because its target table is excluded..** |

System action: Processing continues.

User response: Fix the problem and try again.

| **ADB7144W** | **The** *<insert1>* **attribute of the implicit table space** *<insert2>* **cannot be altered to retain the specification from** *<insert3>* **because of** *<insert4>*. |

Explanation: When either of the tables being compared uses implicit table space, Object Comparison Tool or Administration Tool compares the DB2 default values against the original values from the source or target in order to preserve the attributes of the implicit table space. ALTER statements are generated for the differences. This message is displayed when the attribute referenced in the message cannot be altered on the DB2 release that Object Comparison Tool or Administration Tool is running on.

System action: Processing continues.

User response: The attribute cannot be altered. To alter the attribute, you must establish a DDL with explicit objects or migrate to a DB2 version that supports the alter.

| **ADB7145I** | **Column** *column_name* **cannot be altered by the ALTER TABLE ALTER COLUMN SET WITH DEFAULT statement. Reason code** *reason_code*. |

Explanation: The table cannot be altered due to DB2 restrictions. The table will be dropped and re-created. See the following list for an explanation of the reason code:

1 The table must not be referenced by a view or a Materialized Query Table (MQT).

2 For LOB columns, only the default for inline LOB columns can be changed. The new default length cannot be greater than the inline length.

System action: Processing continues.

User response: No action is required.

| **ADB7146I** | **Table** *table_name* **will be recreated in IBM DB2 Analytics Accelerator.** |

Explanation: This message is issued when a table has been modified and will be offloaded to the IBM DB2 Analytics Accelerator to improve performance.

System action: Processing continues.

User response: No action is required.

| **ADB7147I** | **The tablespace** *table_space* **is a LOB tablespace, which can be dropped only after the auxiliary table has been dropped.** |

Explanation: A comparison of the LOB table spaces shows that the LOB table space needs to be dropped. Object Comparison tool generates the drop statement only when the auxiliary table is dropped because a LOB table space cannot be dropped when an association exists between it and an auxiliary table.

System action: Processing continues.

User response: If necessary, fix the problem and try again.

| **ADB7148I** | **Tablespace** *table_space* **is a LOB tablespace. Because the KEEPTGT option was specified, the tablespace will be kept even if it is not associated with an auxiliary table.** |

Explanation: When the KEEPTGT option is specified, Object Comparison tool will keep the LOB table space which exists on the target but not on the source. Even if the LOB table space is not associated with any auxiliary table after the changes are applied, the LOB table space is still kept.

System action: Processing continues.

User response: If necessary, fix the problem and try again.

ADB7149E **The table:** *table.table* **is partitioned and cannot be dropped explicitly. You can drop the table by dropping the table space** *table.space***.**

Explanation: Because the table space is excluded from the compare process, the table space cannot be dropped. Object Compare is terminated.

System action: Processing stops.

User response: Remove the exclusion on the table space and try the operation again.

ADB7150E *statement table_name* **to** *subsystem* **with privileges** *privileges* **is specified in** *target_name* **exclude specification. This grant is excluded.**

Explanation: ?

System action: Processing continues.

User response: ?

ADB7151W *statement table_name* **to** *subsystem* **with privileges** *privileges* **is specified in** *target_name* **exclude specification. Excluding target grants has no effect because Object Comparison tool always attempts to retain the target grants.**

Explanation: ?

System action: Processing continues.

User response: ?

ADB7154W **The dependent object information is needed. The version files must have a release marker of at least 814. Generate new version files with the current product JCL.**

Explanation: This message is issued if the Object Compare processing requires object dependency information and at least one function, stored procedure, or trigger. It is used to determine the order these objects must be dropped and there is no change to dependency. If either the source or target is a version file with object dependency information, then the object dependency information from that version file will be used.

System action: No system action is taken.

User response: If a version file is not at or above the 814 level, it is recommended that the version file be regenerated using product JCL at the current level.

ADB7155E **The name** *<consname>* **of the** *<constype>* **constraint on the** *<tbfname>* **table is a duplicate name of another** *<dupctype>* **constraint which was previously specified on the same table.**

Explanation: The constraint name must be different from the names of any referential, check, primary key, or unique key constraints previously specified on the table.

System action: Processing stops.

User response: Fix the problem by removing the duplicate name from the constraint definition or renaming the constraint with an unique name, and then try again.

ADB7156E **The** *<insert1>* **tablespace has more than one table. Changing the tablespace to a** *<insert2>* **tablespace will fail. Number of tables is:** *<insert3>*

System action: Processing stops.

User response: Fix the problem and try again.

ADB7158E **The user-defined function** *<function_name>* **from** *<origin>* **is a** *<function_type>***.**

Explanation: This message is displayed when the compared objects include the non-inline SQL scalar function or the SQL table function, and the compare option **Bypass SQL PL functions** (parameter **BYPASSSQLPL**) is not set to Yes.

System action: Processing stops.

User response: Remove the reported SQL PL function from the source or the target and try again. Alternatively, you can set the compare option **Bypass SQL PL functions** to YES and try again.

ADB7157W **The** *<insert1>* **tablespace has more than one table. Changing the tablespace to a** *<insert2>* **tablespace may fail. Number of tables is:** *<insert3>*

System action: Processing continues.

User response: Fix the problem and try again.

ADB7158I **One or more attributes of the implicit** *<obj_type>* **for table** *<tbname> <colname> <part>* **are altered to retain the** *<srctgt>* **value.**

Explanation: When a table is dropped and re-created or is added to the target system, DB2 creates implicit table spaces and indexes for the table with attributes that have default values. ALTER statements are generated to change the default values to the original

target values or to the values from the source objects.

System action: Processing continues.

User response: No action is required.

ADB7159I **Field** *<field_name>* **changed from** *<tgt_value>* **to** *<src_value>* **but no ALTER statement is generated because the new value is the same as the default value from target system.**

Explanation: No ALTER statement is generated because the implicit object is dropped and re-created with the attributes that have a default value that matches the value on the source system.

System action: Processing continues.

User response: No action is required.

ADB7160W **The** *table_name* **table was not placed in read-only mode before the** *trigger_name* **dependent trigger was dropped because the database or table space name of the table is unknown. Data integrity issues can occur during the change of the trigger.**

Explanation: The database name or table space name is unknown for one or more of the following reasons:

- The target is from DDL, and the table on which the trigger is built is not included in the compared objects.
- The target is from DDL, and the table on which the trigger is built is included in the compared objects. However, the table is created in an implicit database and table space.
- A previously created version file, which does not contain database and table space information for the trigger, is being used as the target.

System action: Processing continues.

User response: Review the message. If necessary, take the appropriate action and resubmit the compare job.

If the target is from a DDL file, ensure that the table definition is included in the DDL file, and that the table is not created in an implicit database and table space. Otherwise, extract the target definition from the DB2 catalog by specifying the target from the DB2 catalog on the Compare panel. If a previously created target version file is being used, regenerate the version file.

ADB7161W **The tablespace uses index-controlled partitioning and has changed from non-large to large. Data in the last partition of the table might be discarded if the partitioning limit key is not set to the highest possible value for an ascending index key column or set to the lowest possible value for a**

descending index key column.

System action: Processing continues.

User response: Review the generated APPLY jobs or WSL before applying the change. If necessary, update the limit key of the last partition to avoid the possibility of discarded data during the LOAD phase.

ADB7162W **The number of auxiliary tables associated with the source table might not be consistent with the number of LOB columns in the source table. Implicit LOB objects are used when auxiliary tables are re-created. After changes are applied, ensure that one auxiliary table exists for each LOB column.**

Explanation: For tables that contain LOB columns, DB2 requires that LOB table spaces and auxiliary tables be created to hold the LOB data. When the base table is non-partitioned, DB2 requires one LOB table space and one auxiliary table be created for each LOB column. Object Comparison Tool checks whether the LOB objects definitions on the source agree with DB2 rules. This message is displayed when an inconsistency is found. Object Comparison Tool will re-create the LOB objects implicitly if the table is re-created. When the source comes from DDL file and implicit LOB objects are used, the version file generated from the DDL file might not contain enough information for Object Compare to determine the correctness of the LOB objects definitions.

System action: Processing continues.

User response: After processing completes, assess whether auxiliary table definitions are missing or if implicit LOB objects have been created. If table definitions are missing, fix the problem and try again.

ADB7164W **The logging attribute for** *<obj_desc>* *<objname_v>* **is unknown because the table space is not included in the compared objects. Ensure that the correct SHRLEVEL option is used for the REORG utility.**

Explanation: SHRLEVEL CHANGE or SHRLEVEL REFERENCE REORG might not be executable on a NOT LOGGED table space because of DB2 restrictions. After the change, when the table space is NOT LOGGED, Object Compare will convert the SHRLEVEL option to a valid value if the user-specified SHRLEVEL is not applicable. This message is displayed when the logging attribute of a table space is unknown because the table space is not included in the compared objects. Ensure the correct SHRLEVEL option is used in the APPLY jobs for the REORG utility.

System action: Processing continues.

User response: Review the message. If necessary, fix the problem and try again.

ADB7163W | The number of auxiliary tables associated with the source table might not be consistent with the number of LOB columns in the source table multiplied by the number of partitions in the table space. Implicit LOB objects are used when the base table is re-created. After changes are applied, ensure that one auxiliary table exists for each LOB column in each partition.

Explanation: For tables that contain LOB columns, DB2 requires that table spaces and auxiliary tables be created to hold the LOB data. When the base table is partitioned, DB2 requires one LOB table space and one auxiliary table be created for each LOB column in each partition. Object Comparison Tool checks whether the LOB objects definitions on the source agree with DB2 rules. Because of apparent inconsistency, Object Comparison Tool re-creates the LOB objects implicitly if the table is re-created. When the source comes from DDL file and implicit LOB objects are used, the version file generated from the DDL file might not contain enough information for Object Compare to determine the correctness of the LOB objects definitions. Check and ensure the correctness of the source DDL file.

System action: Processing continues.

User response: After processing completes, assess whether auxiliary table definitions are missing or if implicit LOB objects have been created. If table definitions are missing, fix the problem and try again.

ADB7165I | ALTER is not allowed by DB2 for this operation because *<reason_v>*.

Explanation: ALTER is not allowed for this change because of DB2 restrictions. The object will be dropped and re-created.

System action: Processing continues.

User response: Review the message to determine the reason ALTER is not allowed.

ADB7166E | The EDITPROC is not valid for this table because of DB2 restrictions.

Explanation: The EDITPROC is not valid because of one of the following reasons: 1) The table contains LOB columns, 2) The table cannot have a ROWID, Identity, SECURITY LABEL or XML column when the WITH ROW ATTRIBUTES option is specified, 3) Column names cannot be more than 18 EBCDIC SBCD characters in length when the WITH ROW ATTRIBUTES option is specified.

System action: Processing stops.

User response: Correct the definitions of the table column.

ADB7167W | The *<attribute>* of the implicit XML table space changed from *<value_1>* to *<value_2>*. However, no change statement will be generated because the ALTER statement cannot apply the change.

Explanation: The attribute cannot be altered on the DB2 version that Object Comparison Tool is running on. The DB2 version is earlier than Version 10 new-function mode.

System action: Processing continues.

User response: You cannot alter the attribute on this version. To alter the attribute, you must migrate to a DB2 version that supports the enhanced ALTER statement.

ADB7168E | The source table space cannot contain the table record length. Specify a larger buffer pool to ensure that the page size is suitable for the table record length and that the table space can contain the record.

Explanation: This change cannot be applied until you choose a proper buffer pool for the table space.

System action: Processing stops.

User response: Choose a proper buffer pool for the table space before applying the changes to the table.

ADB7169W | The page size of the table space is unknown because the table space is not included in the compared objects. Ensure that the row length for the table does not exceed the page size limit.

Explanation: Object Compare checks that the row length of the table does not exceed the page size limit. This message is displayed when Object Compare cannot determine the page size of the table space because the table space is not included in compared objects.

System action: Processing continues.

User response: Review the message. Ensure that the table space is specified in compared objects. If necessary, specify a buffer pool with proper page size before running the apply jobs.

ADB7170I | Partitioning changes are ignored. LOB objects related to *<insert1>* partitions are not *<insert2>*.

Explanation: The general ignore option PARTITIONING was specified for this compare run

therefore all changes related to partitioning are ignored. Explicit LOB objects for added partitions are not created. Explicit LOB objects for dropped partitions are not dropped.

System action: Processing continues.

User response: No action is required.

ADB7171W **The source contains an incomplete set of explicit LOB objects therefore all LOB objects for this base table will be created implicitly.**

Explanation: For tables that contain LOB columns, DB2 requires that LOB table spaces, auxiliary tables, and their indexes be created to contain the LOB data. When the base table is created all LOB objects must be created either explicitly or implicitly. Because the source contains explicit definitions for some of the LOB objects, and is missing the definition of other LOB objects, Object Compare creates implicitly all LOB objects for this base table.

System action: Processing continues.

User response: No action is required. However, if you want all LOB objects to be explicit, add the missing definitions and run compare again.

ADB7172W **A single partition or multiple partitions were added by altering the table. New LOB objects for added partitions are created implicitly by DB2. Any explicit definitions of new LOB objects are ignored.**

Explanation: If partitions are added to a table using the ALTER TABLE ADD PARTITION statement and the table is in a PBG table space, then DB2 creates all needed LOB objects (LOB table space, auxiliary table, index on auxiliary table) for the partitions implicitly. Therefore explicit definitions for the LOB objects specified on the source can not be used.

System action: Processing continues.

User response: No action is required.

ADB7173I **The SEGSIZE will be set to the default value 32 after a change of MAXPARTITIONS is applied.**

Explanation: While changes to MAXPARTITIONS are still pending, the SEGSIZE of a simple or segmented table space will not change. Once changes to MAXPARTITIONS are applied, the SEGSIZE will be set to the default value 32 by DB2. Therefore, Compare will not generate a statement for changing SEGSIZE to 32.

System action: Processing continues.

User response: No action is required

ADB7174W **Archive table** *<insert1>*. *<insert2>* **is specified in** *<insert3>* **exclude specification.**

System action: Processing continues.

User response: No action is required.

ADB7175W **Archive-enabled table** *<insert1>*. *<insert2>* **and archive table are both excluded.**

System action: Processing continues.

User response: No action is required.

ADB7176E **Synonym** *syn_name* **for** *syn_creator* **is also found as a** *obj_type*.

Explanation: The name of the synonym has already been found as another *obj_type* in the target.

System action: Processing stops.

User response: Refer to the compare report to correct this error and rerun the job.

ADB7176I **The table will be loaded to IBM DB2 Analytics Accelerator.**

Explanation: This message is issued when a table has been created and its data will be loaded to IBM DB2 Analytics Accelerator

System action: Processing continues.

User response: None.

ADB7177E *Obj_type obj_creator.obj_name* **is also found as a** *obj_type*.

Explanation: *Obj_type obj_creator.obj_name* has already been found as another *obj_type* in the target.

System action: Processing stops.

User response: Refer to the compare report to correct this error and rerun the job.

ADB7177I **The table will be removed from IBM DB2 Analytics Accelerator.**

Explanation: This message is issued when a table has been dropped and will be removed from the IBM DB2 Analytics Accelerator.

System action: Processing continues.

User response: None.

ADB7180E **User Defined SQL Scalar Function**
<insert1> **has versions with different**
SECURED options.

Explanation: All versions of a SQL Scalar function
must be all SECURED or all NOT SECURED.

System action: Processing continues. Object Compare
ends with RC=8.

User response: Make all versions of the procedure
consistent and try again.

ADB7181E **Native Stored Procedure** *procedure* **has**
versions with different COMMIT ON
RETURN options.

Explanation: The versions of a Native Stored
Procedure must be all COMMIT ON RETURN or
AUTONOMOUS.

System action: Processing continues. Object Compare
ends with RC=8.

User response: Make all versions of the procedure
consistent and try again.

ADB7182E **Source Procedure** *<insert1>* **type is**
<insert2> **and target procedure** *<insert3>*
type is *<insert4>*. **To compare native**
stored procedures, both source and
target procedures must be the same
type.

Explanation: Native stored procedures are compared
only when both source and target are of the same type.

System action: Processing continues. Object Compare
ends with RC=8.

User response: Correct source and target procedures
so that the procedure type is the same and then try
again.

ADB7183E **OMPRESS is specified as YES, therefore**
the index is changed to use index
compression. The buffer pool *<insert1>*
must be 8 KB, 16 KB, or 32 KB in size.

Explanation: If compress is changed to YES, then the
size of buffer pool must be 8K, 16K or 32K.

System action: Processing continues.

User response: Make sure the buffer pool size is
correct.

ADB7184E **A duplicate record was detected for**
object *<object name>*. **The error is**
probably caused by the renaming of an
object or by masking from *<target object*
name> **to** *<source object name>*. **The run**
will terminate.

Explanation: A duplicate record error occurred. The
error was probably caused by an attempt to rename an
object or to mask.

System action: Processing stops. A return code of 12 is
generated for the Object Compare step.

User response: Change the name value so that the
rename or mask is no longer a duplicate. Then, try run
the job again.

ADB7185W **User-defined SQL Scalar Function**
<insert1> **has versions with the same**
name but different signatures. This may
cause problems when changes are
applied.

System action: Processing continues. Object Compare
ends with RC=4.

User response: Make all versions of the function
consistent and try again.

ADB7186I **Column** *<colname_v>* **is referenced by**
triggers. The column can not be altered.

System action: Processing continues.

ADB7187E **The version level** *version_level* **in the**
source_target **version file is not**
supported.

Explanation: An unsupported version was detected in
the version file. The version file was created by a prior
release and is not supported.

System action: Return code 8 is set and processing is
halted.

User response: Examine the version in the version file.
Create the version file again using the current release,
or convert the version file to the current release.

ADB7188W **The** *<source_target>* **version file has an**
unknown version level.

Explanation: An unknown version was detected in the
version file. The version file was created by a prior
release.

System action: Return code 4 is set and processing
continues.

User response: Verify that the unknown base version
record is valid.

ADB7190I **Trigger will be dropped and re-created**
because of change to referenced
column(s).

System action: Processing continues.

ADB7191I **Column** column_name **cannot be dropped by the ALTER TABLE DROP COLUMN RESTRICT statement. Reason code:** reason_code**.**

Explanation: You are attempting to drop a column that cannot be dropped. See the following list for an explanation of the reason code:

1 The column is the only column in the table.

2 The column has a security label defined.

3 The column is a DOCID column.

4 The column is a hidden ROWID column.

5 The column is a ROWID column, and a LOB column is dependent on it.

6 The column is defined as ROWID GENERATED BY DEFAULT, and the table contains a hidden ROWID column.

7 The column is part of the table partitioning key.

8 The column is part of the hash key.

9 The remaining columns in the table are all hidden.

10 The column is referenced in the definition of a period.

11 The column is an XML column.

12 The column is referenced by views, indexes, triggers, row permission, column mask, or inline SQL_table functions.

13 The column contains check constraints.

14 The column contains unique constraints.

15 The column contains referential constraints.

System action: Processing continues.

User response: Remove the column restriction or dependency and try again.

ADB7192I **ALTER TABLE DROP COLUMN RESTRICT statement cannot be generated for table** table_name**. Reason code** reason_code**.**

Explanation: You are attempting to drop a column that cannot be dropped. See the following list for an explanation of the reason code:

1 There are triggers defined on the table.

2 The table space is not a universal table space (UTS).

3 The table is a system-period temporal table.

4 The table contains check constraints.

5 The table is a created global temporary table.

6 The table is a history table.

7 The table has an edit procedure or validation exit procedure.

8 The table is referenced by extended indexes, materialized query tables, or inline SQL table functions dependency.

9 The table is referenced by row permissions or column masks.

10 There are INSTEAD OF triggers defined on a view that is dependent on the table.

System action: Processing continues.

User response: Remove the table or tablespace restriction or dependency and try again.

ADB7193I **(PC)** <insert1> **column** <colname> **dropped.**

Explanation: There is an outstanding pending change to drop specific column. The column will be processed as if it was dropped.

System action: Processing continues.

ADB7199E base_type base_name **cannot be dropped because** dep_type dep_name **depends on it.**

Explanation: An object cannot be dropped when another object depends on it. The dependent object cannot be dropped because it is not included in the target. All objects that depend on another object must be in the target so that they can be dropped and re-created if required. Any attempt to drop an object with dependents is rejected by DB2.

System action: Processing is halted and return code 12 is set for the step.

User response: Include all dependent objects in the target. If you are using the ALT command to alter a table, use the ADD primary command from panel ADB27CA and add the objects that are referenced in this message. See *Altering or redefining a table with the ALT command* in the *DB2 Administration Tool for z/OS User's Guide and Reference*.

ADB7200I action_indicator old_option **changed to** new_option**.**

Explanation: The option has been changed. The action_indicator is one of the following:

• (A) - ALTER; the change will be implemented by the ALTER statement.

• (D) - DROP; the change will be implemented by dropping and recreating the object.

System action: Processing continues.

User response: No action is required.

ADB7201I *action_indicator option* **changed from** *target_option* **to** *source_option*

Explanation: The option has been changed. The *action_indicator* is one of the following:

- (A) - ALTER; the change will be implemented by the ALTER statement.
- (D) - DROP; the change will be implemented by dropping and recreating the object.

System action: Processing continues.

User response: No action is required.

ADB7202I *action_indicator* **Default text** *action*: *default_text*

Explanation: The option has been changed. The *action_indicator* is one of the following:

- (A) - ALTER; the change will be implemented by the ALTER statement.
- (D) - DROP; the change will be implemented by dropping and recreating the object.

Action is one of the following:

- added
- deleted

System action: Processing continues.

User response: No action is required.

ADB7203I **Grant(source):** **Grantor=**source_grantor_role source_grantor **Grantee:**source_grantee_role source_grantee **(Not propagated)**

Explanation: If CMDELTA mode is not being used and if a compared object has new grants on the source that are not in the target, Object Comparison Tool will not propagate new grants from the source and will not generate any new source grant statements.

System action: Processing continues.

User response: No action is required.

ADB7204I **Grant(target): Grantor=**target_grantor_role target_grantor **Grantee:**target_grantee_role target_grantee **(Kept)**

Explanation: If the grants exist on the target when the object is dropped and re-created, the target grants are kept.

System action: Processing continues.

User response: No action is required.

ADB7205E **The ALTER TABLE ADD VERSIONING statement cannot be processed, because the history table** *history_table_qualifer.history_table_name* **was not defined at the time the ADD VERSIONING statement was issued in the DDL file.**

Explanation: The specified history table must exist before the ALTER TABLE ADD VERSIONING statement is issued.

System action: Processing stops.

User response: Correct the DDL. Make sure that the history table is defined before the ALTER TABLE ADD VERSIONING statement is issued.

ADB7206I **Grant(target): The subtype for character string columns (column type CHAR, VARCHAR, or CLOB) will be changed from SBCS to MIXED because the encoding scheme of the table is converted to UNICODE.**

Explanation: Character data (CHAR, VARCHAR, and CLOB) is encoded in Unicode UTF-8, which DB2 considers to be mixed data by default.

System action: Processing continues.

User response: None.

ADB7206E **The column definition includes a CCSID attribute that can be specified only if the table has the EBCDIC encoding scheme.**

Explanation: The column attribute CCSID 1208 or CCSID 1200 was specified for a column in a table with an encoding scheme that is not EBCDIC. In such cases, Object compare issues an error message to correct the problem to avoid run time failure.

System action: Processing stops

User response: Correct the encoding scheme for tables with EBCDIC to specify the CCSID attribute in the column definition. After the corrections are made, resubmit the job.

ADB7207E **The column definition includes a CCSID attribute that is not allowed on a table that has EDITPROC or VALIDPROC defined on it.**

Explanation: The column attribute CCSID 1208 or CCSID 1200 was specified for a column in a table with EDITPROC or VALIDPROC defined on it. In such cases, Object Compare issues an error message to correct the problem to avoid run time failure.

System action: Processing stops.

User response: Correct the EDITPROC or VALIDPROC for tables to specify CCSID attribute in column definition. After the corrections are made, resubmit the job.

ADB7208E **The column definition includes a CCSID clause and a FIELDPROC clause. Both clauses are mutually exclusive and are not allowed in the same column definition.**

Explanation: The column attribute CCSID 1208 or CCSID 1200 was specified for a column in a table with the FIELDPROC clause.

System action: Processing stops.

User response: Correct the FIELDPROC clause in the column definition to specify the CCSID attribute in the same column definition. After the corrections are made, resubmit the job.

ADB7205E **The column definition includes a CCSID attribute that can be specified only if the table has the EBCDIC encoding scheme.**

Explanation: The column attribute CCSID 1208 or CCSID 1200 was specified for a column in a table with an encoding scheme that is not EBCDIC. In such cases, Object Compare issues an error message to correct the problem and avoid runtime failure.

System action: Processing stops.

User response: Correct the encoding scheme for tables to EBCDIC to specify the CCSID attribute in column definition. After the corrections are made, resubmit the job.

ADB7210I **The START and RESTART WITH target values will not be changed because 'YES' was specified for the option 'Retain START and RESTART values for sequence object'.**

Explanation: If 'YES' is specified, the START value and RESTART WITH values of the target sequence will be retained and no ALTER SEQUENCE... RESTART statement will be generated. If 'NO' is specified and ignores for START and RESTART fields are not specified, the statement will be generated with values from source to make the target same as the source.

System action: Processing continues.

User response: No action is required.

ADB7221W **The default value for appended column *<column_name>* could not be propagated from the temporal table to the history table.**

Explanation: The default value could not be propagated to the history table when adding a new column in the temporal table because DB2 for z/OS does not allow it.

System action: Processing continues.

User response: None

ADB7222W **The difference in the default value for column *<column_name>* cannot be implemented in a history table.**

Explanation: The default value cannot be altered in the history table because DB2 for z/OS does not allow it.

System action: Processing continues.

User response: None

ADB7350E *<insert1>* **detected an ONCODE condition** *<ONCODE_value>* **in** *<internal_routine>* **at** *<line_number>*.

Explanation: Internal error caused in location in specified module.

System action: Processing stops.

User response: Contact IBM Software Support.

ADB7380E **Module** *module_name* **- Severe error.** *program_name* **has been stopped.**

Explanation: The Object Comparison tool has issued an error message for a severe problem.

System action: A return code of 12 is set and processing stops.

User response: Refer to other error messages generated in the same report for more information on the cause of this error and actions you can take.

ADB7401E **Compressed catalog record failed to decompress.**

Explanation: An error occurred while decompressing the compressed catalog record.

System action: Processing stops.

User response: Contact IBM Software Support.

ADB7402E **Unexpected record type found on** *<insert1>* **file. Expected:** *<insert2>*. **Found:** *<insert3>*.

Explanation: An unexpected record type has been found on source file or target file

System action: Processing stops.

User response: Contact IBM Software Support.

ADB7403E *<insert1>* **limit reached. Max =** *<insert2>.*

Explanation: An error occurred when the number of elements in an array created for relations or user-defined functions reached the maximum limit.

System action: Processing stops.

User response: Contact IBM Software Support.

ADB7404E **Number of version file records generated for an object exceeds the limit.**

Explanation: Too many version file records have been generated for an object.

System action: Processing stops.

User response: Contact IBM Software Support.

ADB7405E **Duplicate drop is detected for object** *<insert1>.*

Explanation: A duplicate explicit drop was detected for an object.

System action: Processing stops.

User response: Contact IBM Software Support.

ADB7406E **Sort Process failed for** *<insert1>* **version file.**

Explanation: An error has occurred during the sorting process of source or target version file records.

System action: Processing stops.

User response: Contact IBM Software Support.

ADB7407E **Unknown catalog record type** *<insert1>.*

Explanation: An unknown record type has been found in the version file.

System action: Processing stops.

User response: Contact IBM Software Support.

ADB7408E **Server error when generating DDL.**

Explanation: An error occurred while generating DDL.

System action: Processing stops.

User response: Contact IBM Software Support.

ADB7701E **The DB2 Admin Tool was unable to load the DB2 DECP module. The return code is** *rc*. **The DDL reader was terminated.**

Explanation: An internal error occurred while loading the DECP module.

System action: A return code was set, and the DDL reader was terminated.

User response: Correct the job by specifying a valid DECP loading action, and resubmit the job. If the problem persists, report this error to IBM Software Support.

ADB7705W **The DDL reader does not support the** *statement_name* **statement.**

Explanation: An unsupported statement is specified in the DDL.

System action: Processing continues with the next statement.

User response: Fix the DDL by correcting the unsupported SQL statement and try again.

ADB7706E **The DDL reader does not support the** *statement_name* **statement.**

Explanation: An unsupported statement is specified in the DDL.

System action: Processing continues with the next statement.

User response: Fix the DDL by correcting the unsupported SQL statement and try again.

ADB7709E **An error occurred while processing the SET PATH statement.**

Explanation: The DDL reader encountered an invalid token in the SET PATH statement.

System action: A return code of 12 is set and processing stops.

User response: Specify a valid SET PATH statement and try again.

ADB7711I **The DDL reader is processing under the authorization ID for the** *userid* **user ID. The authorization ID can be changed by the SET CURRENT SQLID statement.**

Explanation: The SQL ID is for informational purposes only.

System action: The DDL reader continues processing.

User response: No action is required.

ADB7713I **The DDL reader is processing under the authorization ID for the** *schema_name* **schema. The authorization ID can be changed by the SET CURRENT SQLID statement.**

Explanation: The schema name is provided for informational purposes only.

System action: The DDL reader continues processing.

User response: No action is required.

ADB7755E *<colname>* **is not a column of table** *<creator>. <name>*.

Explanation: This error message is displayed if an invalid column name is specified.

System action: A return code of 12 is set, and DTC continues processing.

User response: Fix the problem and try again.

ADB7757E **The following element that is specified for an IDENTITY column is not valid:** *token_name*.

Explanation: While processing an AS IDENTITY clause, the DDL reader encountered an invalid token in the data type expression.

System action: This SQL statement cannot be executed. A return code of 16 is set and processing stops.

User response: Correct the DDL with valid statements, and resubmit the job.

ADB7763I **Temporary (TEMP) databases are not supported by DB2 V9 or higher.**

Explanation: The DDL has a TEMP DATABASES statement, which is not supported by DB2 V9 or higher.

System action: The DDL reader continues processing.

User response: No action is required.

ADB7715E **The DDL reader could not parse a DDL statement. The return code is** *rc*. **The error statement is** *error_stmt*.

Explanation: The specified statement, *error_stmt*, is an invalid SQL statement.

System action: The DDL reader stops processing.

User response: Correct the DDL with valid a SQL statement, and resubmit the job.

ADB7719W **No action taken for the GRANT on** *type* **statement. Processing continues with the next statement.**

Explanation: A GRANT statement specified in either a package or a plan was ignored.

System action: The DDL reader continues processing the next statement.

User response: No action is required.

ADB7723E **There is an unexpected token in the table definition. The token is** *token_name*.

Explanation: The token in the table definition is not valid. The SQL statement cannot be executed.

System action: A return code of 16 is set and processing stops.

User response: Correct the DDL and resubmit the job.

ADB7725E **There is an unexpected token in the table definition. The token is** *token_name*.

Explanation: The token in the table definition is not valid. The SQL statement cannot be executed.

System action: A return code of 16 is set and processing stops.

User response: Correct the DDL and resubmit the job.

ADB7727E **An unexpected token was found in the parameter declaration. The token is** *token_name*.

Explanation: The specified *token_name* is not a valid UDF parameter name. The SQL statement cannot be executed.

System action: A return code of 16 is set and processing stops.

User response: Correct the DDL and resubmit the job.

ADB7729E **An unexpected token was found in the RETURNS clause. The token is** *token_name*.

Explanation: The specified *token_name* is not a valid UDF parameter name. The SQL statement cannot be executed.

System action: A return code of 16 is set and processing stops.

User response: Correct the DDL and resubmit the job.

ADB7731W **A function option was specified that is not valid. The** *token_name* **token was found in the following position:** *processing_position*.

Explanation: The DDL reader encountered a token where it expected to find a function option.

System action: The DDL reader continues processing.

User response: Correct the option name and resubmit the job.

ADB7733E **A function option was specified that is not valid. The** *token_name* **token index is out of range.**

Explanation: The DDL reader encountered an invalid *token_name* as a function option. The SQL statement cannot be executed.

System action: The DDL reader continues processing.

User response: Correct the DDL and resubmit the job. If the problem persists, report this error to IBM Software Support.

ADB7735E **The following element that is specified in an ALTER SEQUENCE statement is not valid:** *token_name*.

Explanation: While processing an ALTER statement, the DDL reader encountered an invalid token in the SEQUENCE expression. The SQL statement cannot be executed.

System action: A return code of 16 is set and processing stops.

User response: Correct the DDL and resubmit the job.

ADB7739E **The DDL reader encountered nested bracketed comments that are not complete.**

Explanation: The DDL reader encountered nested brackets that are not matched. The SQL statement cannot be executed.

System action: A return code of 8 is set and processing stops.

User response: Ensure every comment has an opening and closing bracket. Correct the DDL and resubmit the job.

ADB7741E **The number of entries in the Token index exceeds the number of tokens.**

Explanation: The DDL reader encountered a mismatch between the token index number and the number of tokens. The SQL statement cannot be executed. This is an internal error.

System action: A return code of 16 is set and processing stops.

User response: Report this error to IBM Software Support.

ADB7743E **The DDL reader encountered an SQL statement that is too long.**

Explanation: This is an internal error. The SQL statement cannot be executed.

System action: A return code of 16 is set and processing stops.

User response: Correct the DDL with valid statements, and resubmit the job. If the problem persists, report this error to IBM Software Support.

ADB7753E **The following string is too long:** *string*.

Explanation: The DDL reader encountered a quoted string that is too long. This is an internal error.

System action: A return code of 12 is set and processing stops.

User response: Report this error to IBM Software Support.

ADB7749E **The DDL reader encountered an unexpected token in the following option:** *option_name*.

Explanation: The specified *option_name* is not a valid option name in the SQL PROCEDURE statement. This SQL statement cannot be executed.

System action: A return code of 16 is set and processing stops.

User response: Correct the option name and resubmit the job.

ADB7751I **The** *token_name* **token was found in the Column list, but it is not valid.**

Explanation: While creating a table, the DDL reader encountered an invalid token in the Column list.

System action: The DDL reader continues processing.

User response: Specify valid SQL statements in the DDL and try again.

ADB7765E **The DDL reader encountered the following invalid token after an IN clause:** *token_name*.

Explanation: While creating a table, the DDL reader encountered an invalid token.

System action: A return code of 16 is set and processing stops.

User response: Fix the DDL with valid SQL statements and try again.

ADB7767I **The DDL reader encountered the following partition number, which is not valid:** *partition*.

Explanation: The DDL reader encountered a partition number that exceeds the number of partitions in the table space.

System action: The DDL reader continues processing.

User response: Specify the valid partition number in the DDL and try again.

ADB7769E Empty parentheses () are not permitted following the FLOAT keyword.

Explanation: The DDL FLOAT keyword needs a numerical expression inside parentheses in order for the floating point expression to be translated.

System action: A return code of 16 is set and processing stops.

User response: Provide a numeric expression in the FLOAT keyword and try again.

ADB7771E The DDL reader encountered a substring outside of a string.

Explanation: This is an internal error caused by an invalid string position.

System action: This SQL statement cannot be executed and processing stops.

User response: Correct the DDL with valid statements, and resubmit the job. If the problem persists, report this error to IBM Software Support.

ADB7773W No action was taken for the ALTER *type* REGENERATE statement. Processing continues with the next statement.

Explanation: The DDL reader encountered an unsupported ALTER *type* REGENERATE statement such as INDEX, MASK, PERMISSION, and PROCEDURE.

System action: The DDL readers continues processing.

User response: Fix the DDL with supported SQL statements and try again.

ADB7775I The DDL reader does not support the *type* statement. Processing continues with the next statement.

Explanation: A ROLE or TRUSTED context is specified in an SQL statement, which is not supported.

System action: The DDL readers continues processing.

User response: Fix the DDL with supported SQL statements and try again.

ADB7776E The length specification of the column *col_name* in table *table_name* is invalid.

Explanation: The length specification of the column is invalid.

System action: Processing stops.

User response: Fix the DDL with supported SQL statements and try again.

ADB7777E The DDL reader encountered a CCSID *ccsid|clause* which is not valid for the *encoding|data* type *type*.

Explanation: The DDL reader encountered a CCSID that is not valid for the data type or encoding type. The DDL reader uses the DB2 encoding scheme to verify CCSID values.

System action: None. DTC continues processing.

User response: Refer to documentation for DB2 Version 11 NFM or later for more information on valid CCSID values. Fix the DDL with a valid CCSID, and try again.

ADB7778E The DDL reader encountered mutually exclusive clauses in the ALTER TABLE statement.

Explanation: You can only specify the same clause once, except for the ADD COLUMN and ALTER COLUMN clauses. The ALTER COLUMN, ADD PARTITION, and ROTATE PARTITION clauses are mutually exclusive.

System action: The DDL reader stops processing.

User response: Fix the DDL with supported SQL statements and try the operation again.

ADB7830E The node with the key *key_name* already exists in the dictionary.

Explanation: The node with the key cannot be inserted into the dictionary because the key already exists. This is an internal error.

System action: Processing stops.

User response: Report this internal error to IBM Software Support.

ADB7900I Version File is at current level: *version_level*. No conversion necessary.

Explanation: The Version File does not need to be converted.

System action: Return code = 0. Processing continues.

User response: No action is required.

ADB7902E Unsupported DB2 release: *DB2 release*

Explanation: The DB2 release from the Version File header is not supported.

System action: Return code = 12. Processing stops.

User response: Recreate a new version file at the current level and then try again.

ADB7904E **Unsupported or invalid version file row type:** *row type*

Explanation: A version file row type is not valid.

System action: Return code = 8. Processing continues.

User response: This is a processing error. Contact IBM Software Support.

ADB7910E **Version File error:** *error code*

Explanation: A Version File error has occurred. The type of error is indicated by the errorcode.

Errorcode = 8: The version file is empty.

Errorcode = 12: The version file is missing or its name is not correct.

System action: The Version File Conversion Tool terminates processing.

User response: Correct the data set name or member name.

ADB7913E **The old and new version files have the same name:** *name*

Explanation: The new version file and the old version file have the same name.

System action: A return code of 8 is set. The Version File Conversion Tool terminates processing the current version file.

User response: Correct the new version file dataset name or member name.

ADB7915E **Invalid combination of parameters.**

Explanation: The following combinations of keyword parameters are valid: VFOLD and VFNEW, VOWNER and VNAME, or VID.

System action: A return code of 8 is set. The Version File Conversion Tool terminates processing the current version file.

User response: Correct the parameters.

ADB7916E **Parameter** *parameter* **specfied,** *missing parm* **omitted. Both are required. The version file is not defined.**

Explanation: You must specify both VFOLD and VFNEW parameters or both VOWNER and VNAME parameters for the Version File conversion tool.

System action: A return code of 8 is set. The Version File Conversion Tool terminates processing the current version file.

User response: Correct the parameters.

ADB7917E **Unable to access** *parameter* **version file** *version_file*

Explanation: The specified version file could not be opened. Correct the file name.

System action: A return code of 8 is set. The Version File Conversion Tool terminates processing the current version file.

User response: Correct the parameters.

ADB7918E **Version id =** *version_ID* **was not found.**

Explanation: The specified version ID was not found in the base version database.

System action: A return code of 8 is set. The Version File Conversion Tool terminates processing the current version file.

User response: Correct the parameters.

ADB7919E **Unable to determine the DB2 version for row type:** *row_type*.

Explanation: The DB2 version for the specified version file row type could not be determined.

System action: A return code of 12 is set. The Version File Conversion Tool terminates processing the current version file.

User response: Contact IBM Software Support.

ADB7923E **The keyword parameter,** *keyword* **is invalid.**

Explanation: The specified keyword parameter is invalid for the Version File Conversion Tool.

System action: A return code of 12 is set. The Version File Conversion Tool terminates processing the current version file.

User response: Correct the keyword parameter and try again.

ADB7928E **The version file has an unknown version level.**

Explanation: An unknown version was detected in the version file. The version file was created by a prior release.

System action: A return code of 8 is set. The Version File Conversion Tool terminates processing the current version file.

User response: Recreate a new version file at the current level and then retry.

ADB7951E **An invalid action was specified for the saved compare results.**

Explanation: The input job that Object Compare generated contains an invalid value for the action to save the compare results. The action for the saved compare results must be either ADD or REPLACE.

System action: A return code of 12 is set, and processing stops.

User response: Edit the input job to specify a valid action, and resubmit the job. Report this internal error to IBM Software Support.

ADB7952W **IBMDB2 Analytics Accelerator is not available for the current DB2 subsystem.**

Explanation: Admin or Object Comparison Tool has detected and reloaded the accelerated tables that contain modified data, but DB2 Analytics Accelerator is not available for the current DB2 subsystem.

System action: A return code of 4 is set, and processing continues.

User response: Turn off Reload accelerated tables on panel ADB2PCO or install DB2 Analytics Accelerator for the current DB2 subsystem.

ADB8001E **The second record in a record pair was not found in the input version file.**

Explanation: During the merge operation, required information was not available.

System action: Processing stops.

User response: This is an internal error. Contact IBM Software Support. Details about the object that caused the error are provided in message ADB8057I.

ADB8002E **An internal error occurred for an unknown row type of *<type>*.**

Explanation: The input version file format is not valid.

System action: Processing stops.

User response: This is an internal error. Contact IBM Software Support. Details about the object that caused the error are provided in message ADB8057I.

ADB8003E **A wildcard character (*) was used to specify a volume name, but an explicit volume name has already been specified.**

Explanation: An error occurred while an ALTER storage group was being processed.

System action: Processing stops.

User response: Review the volume name, correct the appropriate statements, and try again. Details about the

object that caused the error are provided in message ADB8057I.

ADB8004E **The volume ID, *volume_id*, cannot be added because a wildcard character (*) was already specified on the storage group.**

Explanation: An error occurred while an ALTER storage group was being processed.

System action: Processing stops.

User response: Review the volume ID, correct the appropriate statements, and try again. Details about the object that caused the error are provided in message ADB8057I.

ADB8005E **The volume ID, *volume_id*, cannot be added to storage group, *obj_name*. The volume is already part of the storage group.**

Explanation: An error occurred while an ALTER storage group was being processed.

System action: Processing stops.

User response: Review the volume ID, correct the appropriate statements, and try again. Details about the object that caused the error are provided in message ADB8057I.

ADB8006W **An attempt was made to drop an *obj_type obj_name* that does not exist.**

Explanation: During the merge operation, an error occurred and the object could not be dropped.

System action: Processing continues.

User response: If necessary, ensure that the object to be dropped is specified correctly, correct the appropriate statements, and try again. Details about the object that caused the error are provided in message ADB8057I.

ADB8007E **An attempt was made to create an *obj_type* that already exists.**

Explanation: During the merge operation, an error occurred and the object was not created.

System action: Processing stops.

User response: Ensure that the object to be created is specified correctly, correct the appropriate statements, and try again. Details about the object that caused the error are provided in message ADB8057I.

ADB8008E   An internal error occurred for an unknown transaction.

Explanation: An error occurred while an object was being processed.

System action: Processing stops.

User response: This is an internal error. Contact IBM Software Support. Details about the object that caused the error are provided in message ADB8057I.

ADB8009E   An attempt was made to rename an *obj_type* **that does not exist.**

Explanation: During the merge operation, an error occurred and the object could not be found and renamed.

System action: Processing stops.

User response: Ensure that the object to be renamed is specified correctly, correct the appropriate statements, and try again. Details about the object that caused the error are provided in message ADB8057I.

ADB8010E   An object cannot be renamed to a specified new name because the new name was already specified in a previous rename operation.

Explanation: An attempt was made to rename an object. The new name was assigned in a previous rename operation and cannot be used for this object.

System action: Processing stops.

User response: Ensure that the rename that was specified is unique and try again. Details about the object that caused the error are provided in message ADB8057I.

ADB8011E   An attempt was made to alter an object that does not contain the record to change.

Explanation: During the merge operation, an error occurred. No object row was found to match a delta row of a specific type.

System action: Processing stops.

User response: Ensure that the object, and particularly the row type, is specified correctly. Correct the appropriate statements and try again. Details about the object that caused the error are provided in message ADB8057I.

ADB8012E   An expected version file record *row_type* **was not found in a base version record.**

Explanation: During the merge operation, an error occurred. A record of a specific row type was expected but was not found.

System action: Processing stops.

User response: This is an internal error. Contact IBM Software Support. Details about the object that caused the error are provided in message ADB8057I.

ADB8013E   An attempt was made to alter a column record, but the specified table does not contain this column *column_name***.**

Explanation: During the merge operation, an error occurred. A column, specified to be updated when altering a table, was not found.

System action: Processing stops.

User response: Ensure that the object, and particularly changes to the column records, is specified correctly. Correct the appropriate statements and try again. Details about the object that caused the error are provided in message ADB8057I.

ADB8014E   An attempt was made to alter the table attributes of a table that does not contain the *rowtype* **record to be changed.**

Explanation: During the merge operation, an error occurred. A column record, of a specific row type and specified to be updated when altering a table, was not found.

System action: Processing stops.

User response: Ensure that the object, and particularly the row type, is specified correctly. Correct the appropriate statements and try again. Details about the object that caused the error are provided in message ADB8057I.

ADB8015E   An attempt was made to change the access control for a table that cannot be found.

Explanation: An error occurred while access to a table row or column was being activated or deactivated.

System action: Processing stops.

User response: Review the access control specified for the table, correct the appropriate statements, and try again. Details about the object that caused the error are provided in message ADB8063I.

ADB8016E   An attempt was made to add or alter the ORGANIZE BY HASH clause for a table, but the corresponding record in the table was not found.

Explanation: During the merge operation, an error occurred. The ORGANIZE BY HASH clause could not be used in a table object.

System action: Processing stops.

User response: Ensure that the table object is specified correctly, correct the appropriate statements, and try again. Details about the object that caused the error are provided in message ADB8057I.

ADB8017E   **An attempt was made to add a SYSTEM_TIME or BUSINESS_PERIOD clause to a table, but the corresponding record in the table was not found.**

Explanation: During the merge operation, an error occurred. A row that was specified to be updated when altering a table was not found.

System action: Processing stops.

User response: Ensure that the table object is specified correctly, correct the appropriate statements, and try again. Details about the object that caused the error are provided in message ADB8057I.

ADB8018E   **An attempt was made to add the** *col_name* **column to the** *table_name* **table, but** *col_name* **already exists in this table.**

Explanation: During the merge operation, an error occurred and the column was not added.

System action: Processing stops.

User response: Ensure that the object to be created is specified correctly, correct the relevant statements, and try again. Details about the object that caused the error are provided in message ADB8057I.

ADB8019E   **An attempt was made to drop an** *obj_type* **that does not exist.**

Explanation: During the merge operation, an error occurred and the object could not be dropped.

System action: Processing stops.

User response: Ensure that the object to be dropped is specified correctly, correct the appropriate statements, and try again. Details about the object that caused the error are provided in message ADB8057I.

ADB8020E   **An attempt was made to add a primary key to the table** *table_name* **, but this table already has a primary key.**

Explanation: An error occurred while a primary key was being added to a table. A table can have only one primary key.

System action: Processing stops.

User response: Ensure that the table and the key to be added are specified correctly, correct the relevant statements, and try again. Details about the object that caused the error are provided in message ADB8057I.

ADB8021E   **An attempt was made to add a primary or unique constraint to the** *table_name* **table, but a constraint with the same name already exists for this table.**

Explanation: An error occurred while a primary or unique constraint was being added to a table.

System action: Processing stops.

User response: Ensure that the table and the constraint to be added are specified correctly, correct the relevant statements, and try again. Details about the object that caused the error are provided in message ADB8057I.

ADB8022E   **An attempt was made to add a primary or unique key, but the column associated with the key,** *col_name* **was not found.**

Explanation: An error occurred while a primary or unique key was being added to a table.

System action: Processing stops.

User response: Ensure that the table column and the key to be added are specified correctly, correct the relevant statements, and try again. Details about the object that caused the error are provided in message ADB8057I.

ADB8023E   **An attempt was made to add the table check constraint** *check_name* **to the table** *table_name*, **but a constraint with the same name already exists for this table.**

Explanation: An error occurred while a table check constraint was being added a table. The same constraint name is already being used as a different check.

System action: Processing stops.

User response: Ensure that the table and the table check constraint to be added are specified correctly, correct the relevant statements, and try again. Details about the object that caused the error are provided in message ADB8057I.

ADB8024E   **The merge process stopped due to severe errors.**

Explanation: The merge process stopped due to severe errors.

System action: Processing stops.

User response: Review other messages that accompany this message to determine the appropriate response.

ADB8025E   An attempt was made to process an invalid add operation for a table.

Explanation: An internal error occurred while processing an ADD operation for a table. The operation type is not valid.

System action: Processing stops.

User response: This is an internal error. Contact IBM Software Support. Details about the object that caused the error are provided in message ADB8057I.

ADB8026E   The ROTATE PARTITION option cannot be processed. Reason code = rc.

Explanation: An error occurred while an ALTER TABLE statement that specifies rotating partitions was being processed. The reason code indicates the source of the error:

1   The table is not partitioned

2   No table partitions exist

3   The row specified for rotate is unknown.

System action: Processing stops.

User response: Review the ALTER TABLE statement that was specified, particularly the ROTATE PARTITION option. Correct the appropriate statements and try again. If the reason code is 3, contact IBM Software Support and provide the information in this message.

ADB8027E   An error occurred in the ADBDICT module: msg.

Explanation: An internal error occurred in a dictionary module.

System action: Processing stops.

User response: This is an internal error. Contact IBM Software Support. Details about the object that caused the error are provided in message ADB8057I.

ADB8028E   An attempt was made to drop a column, but that column does not exist in the obj_type.

Explanation: During the merge operation, an error occurred and the column was not removed from the object.

System action: Processing stops.

User response: Ensure that the object to be dropped is specified correctly, correct the appropriate statements, and try again. Details about the object that caused the error are provided in message ADB8057I.

ADB8029E   An attempt was made to drop the check constraint const_name, but that constraint does not exist in the table table_name.

Explanation: An error occurred while a constraint was being dropped from a table.

System action: Processing stops.

User response: Ensure that the table and the constraint to be dropped are specified correctly, correct the relevant statements, and try again. Details about the object that caused the error are provided in message ADB8057I.

ADB8030E   An attempt was made to process an invalid drop operation for a table.

Explanation: An internal error occurred while processing a DROP operation for a table.

System action: Processing stops.

User response: This is an internal error. Contact IBM Software Support. Details about the object that caused the error are provided in message ADB8057I.

ADB8031E   An attempt was made to insert a column col_name in a table table_name, but col_name already exists in this table.

Explanation: An error occurred while a column was being inserted into a table. The column already exists.

System action: Processing stops.

User response: Ensure that the object to be inserted is specified correctly, correct the relevant statements, and try again. Details about the object that caused the error are provided in message ADB8057I.

ADB8032E   An attempt was made to insert a column col_name, but the specified position was not found.

Explanation: An error occurred while a column was being inserted into a table. During the merge operation, the position for column was determined to be invalid.

System action: Processing stops.

User response: Ensure that the column is specified correctly, correct the appropriate statements, and try again. Details about the object that caused the error are provided in message ADB8057I.

ADB8033E   An internal error occurred. The table was not in the dictionary.

Explanation: An internal error occurred during the renaming of a table.

System action: Processing stops.

User response: This is an internal error. Contact IBM

Software Support. Details about the object that caused the error are provided in message ADB8057I.

ADB8034E   COMMENT ON or LABEL ON on a column for the VIEW *obj_name* **cannot be processed. Column** *col_name* **is not in the view.**

Explanation: The comment or label on a statement is ignored because the column was not found in the view.

System action: Processing continues.

User response: Ensure that the column is specified correctly, correct the appropriate statements, and run the job again.

ADB8035E   An attempt was made to update a version file row, but the matching row specified in a delta change was not found.

Explanation: An internal error occurred while an object was being altered.

System action: Processing stops.

User response: This is an internal error. Contact IBM Software Support. Details about the object that caused the error are provided in message ADB8063I.

ADB8036E   MERGE could not find and update the *obj_type* **version.**

Explanation: An internal error occurred while processing an ALTER FUNCTION or ALTER PROCEDURE statement. During the merge operation, the version of the stored procedure or function was not found.

System action: Processing stops.

User response: Refer to message ADB8057I to determine the stored procedure or function that could not be found and then review the specified ALTER FUNCTION or ALTER PROCEDURE statement that was specified.

ADB8037E   An attempt was made to replace a function with version *ver_id*, **but that version does not exist.**

Explanation: An error occurred while processing an ALTER function statement. During the merge operation, the specified version of the function was not found.

System action: Processing stops.

User response: Refer to message ADB8057I to determine the specific function and then review the specified ALTER FUNCTION statement.

ADB8038I   A DDL statement could not be parsed. Processing continues. RC = *<return_code>*.

Explanation: An error occurred while processing a DDL statement of an object. The reason code indicates the source of the error:

1   An error occurred for a view object.

2   An error occurred for a RENAME statement.

System action: Processing continues.

User response: Correct the DDL statement, if necessary, and run the job again.

ADB8039E   MERGE encountered an error while registering an object. The *obj_type* **already exists with the same name of** *obj_name*.

Explanation: An error occurred while an object was being registered. An object with the same object name already exists.

System action: Processing stops.

User response: Ensure that the object is specified correctly, correct the appropriate statements, and try again. Details about the object that caused the error are provided in message ADB8063I.

ADB8040E   An error occurred during sort processing of the *vf_type* **file: Return code from SORT =** *return_code*.

Explanation: An internal sort process resulted in an error.

System action: Processing stops.

User response: This is an internal error. Contact IBM Software Support and provide the return code and the information in message ADB8057I.

ADB8041W   Dropped foreign key *key_name* **for table** *obj_name* **does not exist. The foreign key might have been dropped when the parent key was dropped.**

Explanation: The specified foreign key does not exist.

System action: Processing continues.

User response: If necessary, ensure the foreign key is specified correctly and run the job again.

ADB8042I   No records were found in the base version file.

Explanation: During the merge operation, no records were found in the base version file.

System action: Processing continues.

User response: Review the base version file. Correct the file, if necessary, and run the job again.

ADB8043I   No delta changes to process.

Explanation: No change records were found in the delta version file.

System action: Processing continues.

User response: Review the change and the delta version file. Correct the file, if necessary, and run the job again.

ADB8044I   No objects to process.

Explanation: No input records were found.

System action: Processing continues.

User response: Review the base and delta version files. Correct the files, if necessary, and run the job again.

ADB8045I   The number of catalog rows exceeds the limit specified for the process.

Explanation: The number of catalog rows exceeds the limit specified for the process.

System action: Processing continues.

User response: This is an internal error. If necessary, contact IBM Software Support.

ADB8046W   The volume *vol_id* **that was specified to be removed was not found in the storage group** *obj_name*.

Explanation: The volume ID to be removed was not found in the storage group.

System action: Processing continues.

User response: If necessary, locate the volume, confirm that removal was specified, and then run the job again.

ADB8047E   KY rows were not found. Alter was attempted for the implicit unique index for table *table_name*.

Explanation: : An internal error occurred during the altering of an implicit index for a table.

System action: Processing stops.

User response: This is an internal error. Contact IBM Software Support. Details about the object that caused the error are provided in message ADB8057I.

ADB8048E   COMMENT ON or LABEL ON on a column for the VIEW *obj_name* **cannot be processed. Column** *col_name* **is not in the view.**

Explanation: : The comment or label on a statement is ignored because the column was not found in the view.

System action: Processing continues.

User response: Ensure that the column is specified correctly, correct the appropriate statements, and run the job again.

ADB8049I   During the ALTER procedure, *obj_type* *obj_name* **was found, but the** *obj_type* **was not found. The** *obj_type* **is assumed to be implicitly created.**

Explanation: : An attempt was made to alter an implicitly created object. Implicitly created objects cannot be altered.

System action: Processing continues.

User response: No response.

ADB8050W   Drop alias *obj_name* **ignored. Alias does not exists.**

Explanation: : A Drop Alias statement is ignored because the alias does not exit.

System action: Processing continues.

User response: : If necessary, correct the change and run the job again.

ADB8051W   Alter found for *obj_name*, **but no object definition was found in base.**

Explanation: : A change was found for an object, but no base definition for the object was found.

System action: Processing continues.

User response: : Ensure that the object is specified correctly, correct the appropriate statements, and run the job again. Details about the object that caused the error are provided in message ADB8057I.

ADB8052E   A change was found for *obj_name*, **but no object definition was found in base.**

Explanation: : A delta change exists for an object that is not defined.

System action: Processing stops.

User response: : Ensure that the object is specified correctly, correct the appropriate statements, and run the job again. Details about the object that caused the error are provided in message ADB8057I.

ADB8053W   A drop was specified for *obj_name*, **but no object definition was found in base.**

Explanation: : An attempt was made to drop an object that is not defined.

System action: Processing continues.

User response: : Ensure that the object is specified correctly, correct the appropriate statements, and run the job again.

ADB8054I   Internal rows, AR or XR, were not found during a search of the LOB or XML entries in the base version file. Row type: *row_type*.

Explanation: : During the merge operation, an expected auxiliary table or XML record was not found in the base records.

System action: Processing continues.

User response: : Review LOB and XML entries. If necessary, correct the statements and run the job again.

ADB8055I   The row type CO was not found in delta stack. No match to the corresponding implicit rows in the base change was found for rowtype: *row_type*.

Explanation: :

An implicit column change, which was flagged as a delta change, was found, but no matching column definition was found. The implicit rows might have been created during internal processing.

.

System action: Processing continues.

User response: : Ensure that the object, and particularly the row type, is specified correctly. If necessary, correct the appropriate statements and run the job again.

ADB8056E   The statement CREATE TABLE *<table>* **LIKE** *<table>* **is not yet supported.**

Explanation: : The statement CREATE TABLE with LIKE predicate is not supported in the merge operation.

System action: Processing stops.

User response: : Remove the statement and try again.

ADB8057I   An error occurred during MERGE processing. The following details apply to the error: Operation: *operation*, **Object name:** *object_name*, **Row type:** *row_type*, **Procedure:** *proc_name*.

Explanation: The message text provides details about

objects and procedures that are involved in the error.

System action: Processing continues.

User response: Use the message text information to correct the problem, or provide the information when you contact IBM Software Support.

ADB8058W   The statement CREATE TABLE *<table>* **LIKE** *<table>* **is not yet supported.**

Explanation: : The statement CREATE TABLE with LIKE predicate is not supported in the merge operation. The statement is ignored.

System action: Processing continues.

User response: : Optionally, remove the statement and try again.

ADB9001W   A parameter name in the input parameter file was not recognized.

Explanation: The input parameter file contains a parameter name that is not valid. The job might not have run correctly because of the incorrect parameter name.

System action: A return code of 4 is set, and processing continues. The program continues on to the next request after reporting the error.

User response: Correct the invalid parameter, and resubmit the job.

ADB9002W   Comments are not allowed in the input parameter file.

Explanation: The input parameter file cannot contain comments. The job might not have run correctly.

System action: A return code of 4 is set, and processing continues. The program continues on to the next request after reporting the error.

User response: Delete the comments from the input parameter file, and resubmit the job.

ADB9003W   Invalid input from the input parameter file is ignored.

Explanation: The input parameter file contains invalid input, which is ignored. The job might not have run correctly because of the invalid input.

System action: A return code of 4 is set, and processing continues. The program continues on to the next request after reporting the error.

User response: Correct the invalid parameter, and resubmit the job.

ADB9004W Processing continues.

Explanation: This message is issued in conjunction with message ADB9001, ADB9002, or ADB9003 to indicate that processing continues when the program encounters these errors.

System action: Processing continues.

User response: None.

ADB9005W The following input was skipped: *error_text*.

Explanation: The job might not have run correctly because input was skipped. *error_text* identifies the input that was skipped.

System action: A return code of 4 is set, and processing continues.

User response: Correct the input, and resubmit the job.

ADB9006I The program *program_name* **completed abnormally.**

Explanation: The accompanying messages indicate why the identified program did not complete normally.

System action: None..

User response: See the accompanying messages in the report.

ADB9007E A version name was not specified.

Explanation: The request cannot be processed because a version name was not specified.

System action: A return code of 12 is set, and processing stops.

User response: Specify a valid version name, and resubmit the request.

ADB9008E A version qualifier was not specified.

Explanation: The request cannot be processed because a version qualifier was not specified.

System action: A return code of 12 is set, and processing stops.

User response: Specify a valid version qualifier and resubmit the request.

ADB9009E Package *module_name* **needs to be bound or rebound.**

Explanation: An SQL statement has been issued, and DB2 has returned an SQLCODE of -805, which indicates that the program needs to be bound or rebound on that particular DB2 system.

System action: A return code of 12 is set, and processing stops.

User response: Bind or rebind the named module, and resubmit the job.

ADB9010E A plan access error occurred for program *program_name* **because you are not authorized to run the plan.**

Explanation: The identified program did not run successfully because the program attempted to issue an SQL request, and DB2 issued an SQLCODE of -922.

System action: A return code of 12 is set, and processing stops.

User response: Correct the authorization, and resubmit the job.

ADB9011E An unexpected sqlcode was found in *error_function*.

Explanation: This message is issued when the environment in which the program is running is not correct or a possible user error exists.

System action: A return code of 12 is set, and processing stops.

User response: Obtain a dump, and contact IBM Software Support.

ADB9012E The DD statement *ddname* **is missing or is incorrect.**

Explanation: The JCL for the job is missing the identified DD statement or the DD statement is incorrect.

System action: A return code of 12 is set, and processing stops.

User response: Supply the missing DD statement, and resubmit the job.

ADB9013E The specified scope *scope_qualifier.scope_name* **was not found.**

Explanation: The request required the use of a version scope and could not be processed because the scope that was specified does not exist.

System action: A return code of 8 is set, and processing stops.

User response: Correct the scope qualifier, scope name, or both to identify a scope that exists, and resubmit the request.

ADB9014I **The specified version** *version_qualifier.version_name* **was found in the database.**

Explanation: The request was processed because the specified version exists.

System action: None.

User response: None.

ADB9015E **The specified version** *version_qualifier.version_name* **was not found in the database.**

Explanation: The request could not be processed because the specified version does not exist.

System action: A return code of 8 is set, and processing stops.

User response: Correct the version qualifier, the version name, or both to identify a version that exists, and resubmit the request.

ADB9016W **The specified version** *version_qualifier.version_name* **exists but its definition is empty or incomplete.**

Explanation: The request might not have been processed accurately because the version is not defined correctly.

System action: A return code of 4 is set, and processing continues.

User response: Correct the version qualifier, the version name, or both and ensure that the version has version records.

ADB9017I *program_name* **- Export Version Files**

Explanation: This report message identifies the DB2 Admin program that is being run to export version files.

System action: None.

User response: None.

ADB9019I **The number of version data records exported is** *integer***.**

Explanation: After the DB2 Admin program to export version files completes, this report message is issued to indicate the number of version file records that were exported.

System action: None.

User response: None.

ADB9020I *ADBCVIC or ADBCVIM* **- Import Version Files**

Explanation: This report message identifies the DB2 Admin program that is being run to import version files.

System action: Processing continues.

User response: None.

ADB9021I **Version Import Complete. The import for version** *ADBCVIC or ADBCVIM* **completed successfully.**

Explanation: This report message indicates that the DB2 Admin program to import version files ran successfully.

System action: None.

User response: None.

ADB9022E **An invalid version type was specified. The valid values are BASE and DELTA.**

Explanation: The input job that DB2 Admin generated contains an invalid value for the version type. The type of version must be either BASE or DELTA.

System action: A return code of 12 is set, and processing stops.

User response: Edit the input job to specify a valid version type, and resubmit the job. Report this internal error to IBM Software Support.

ADB9023E **An invalid action for a version file import was specified. The valid values are ADD and REPLACE.**

Explanation: The input job that DB2 Admin generated contains an invalid value for the action to take when importing a version file. The action for the import must be either ADD or REPLACE.

System action: A return code of 12 is set, and processing stops.

User response: Edit the input job to specify a valid action, and resubmit the job. Report this internal error to IBM Software Support.

ADB9024I **The scope ID for version scope** *scope_qualifier.scope_name* **is scope** *scope_identifier***.**

Explanation: This report message indicates that the version scope with the identified scope ID is being processed.

System action: Processing continues.

User response: None.

ADB9025I **The version will be replaced.**

Explanation: A version is being created. A version with the specified name already exists and will be overwritten.

System action: Processing continues.

User response: None.

ADB9026E **The version already exists. It cannot be added.**

Explanation: DB2 Admin is trying to process a request to add a new version. The version cannot be created because a version with the specified qualifier and name already exists.

System action: A return code of 8 is set, and processing stops.

User response: Report this internal error to IBM Software Support.

ADB9027E **The input file is empty. No records were found.**

Explanation: DB2 Admin is trying to process a request but the input file that describes the action that should be taken is empty.

System action: A return code of 12 is set, and processing stops.

User response: Report this internal error to IBM Software Support.

ADB9028I **A version file was created from** *DB2_subsystem_id* **at** *extract_time* **by** *extract_sqlid*.

Explanation: This report message provides information about the version file that is being processed. It displays the ID of the DB2 subsystem, the time the version file was extracted, and the ID of the user who ran the DB2 Admin program to extract the version information.

System action: None.

User response: None.

ADB9029I **A version file was extracted from location** *DB2_location* **at** *extract_time* **by** *extract_sqlid*.

Explanation: This report message provides information about the version file that is being processed. It displays the location of the DB2 subsystem, the time the version file was extracted, and the ID of the user who ran the DB2 Admin program to extract the version information.

System action: None.

User response: None.

ADB9030E **The version file description is not available because the input file does not have a header record.**

Explanation: DB2 Admin is trying to process a version file but cannot because the input file does not have a header record.

System action: A return code of 12 is set, and processing stops.

User response: Report this internal error to IBM Software Support.

ADB9031W **The input file is empty. No records were found.**

Explanation: DB2 Admin is trying to process a request but cannot because the input file is empty.

System action: A return code of 4 is set, and processing continues.

User response: Report this internal error to IBM Software Support.

ADB9302E **Change "***change_owner.change_name***" cannot be recovered because the following changes must be recovered first and either they do not have a recover change or they have a recover change that is not in the ANALYZED state. Owner.Name**
--

Explanation: An attempt is being made to recover a change that cannot be recovered because other changes must be recovered first and those changes either do not have a recover change or have a recover change that is not in the ANALYZED state. The accompanying messages provide a list of the changes that must be recovered first that either do not have a recover change or have a recover change that is not in the ANALYZED state.

System action: Processing stops.

User response: Create a new change to undo the changes for the specified changes.

ADB9032I **The number of version data records imported is** *integer*.

Explanation: After the DB2 Admin program to import version files completes, this report message is issued to indicate the number of version data records that were exported.

System action: None.

User response: None.

ADB9033I   **The SQLCA sqlcode is** *sqlca.sqlcode***.**

Explanation: This message displays the SQLCODE that was returned.

System action: None.

User response: None.

ADB9034I   **ADBCVOB - Object Extraction Complete**

Explanation: This report message indicates that the DB2 Admin program to extract objects completed successfully.

System action: None.

User response: None.

ADB9035I   **he number of objects that were found is** *integer***.**

Explanation: After the DB2 Admin program to extract objects completes, this report message is issued to indicate the number of objects that were processed.

System action: None.

User response: None.

ADB9036I   **ADBCVOB - Extract Version Objects.**

Explanation: This report message indicates that the DB2 Admin program that extracts the objects for a version has started.

System action: Processing continues.

User response: None.

ADB9037I   **ADBCVSX - Export Scope Objects**

Explanation: This report message indicates that the DB2 Admin program that extracts version scopes has started.

System action: Processing continues.

User response: None.

ADB9038I   **ADBCVSX - Scope Export Complete**

Explanation: This report message indicates that the DB2 Admin program that extracts version scopes has completed successfully.

System action: None.

User response: None.

ADB9039E   **A scope name was not specified**

Explanation: The DB2 Admin program to extract a version scope could not run because the input to the program did not include the scope name.

System action: A return code of 12 is set, and processing stops.

User response: Report this internal error to IBM Software Support.

ADB9040E   **A scope qualifier was not specified.**

Explanation: The DB2 Admin program to extract a version scope could not run because the input to the program did not include the qualifier for the scope.

System action: A return code of 12 is set, and processing stops.

User response: Report this internal error to IBM Software Support.

ADB9041I   **The scope** *scope_qualifier.scope_name* **was found in the database. Its scope ID is** *scope_identifier***.**

Explanation: The scope that was being processed was found, and it has the identified scope ID.

System action: Processing continues.

User response: None.

ADB9042I   **The number of scope objects written is** *integer***.**

Explanation: After the DB2 Admin program to process scope objects completes, this report message is issued to indicate the number of scope objects that were processed.

System action: None.

User response: None.

ADB9043I   **Its scope ID is** *scope_identifier***.**

Explanation: A version scope with the identified scope identifier is being processed.

System action: Processing continues.

User response: None.

ADB9044I   **The version will be added.**

Explanation: The DB2 Admin program that processes versions will add a version.

System action: Processing continues.

User response: None.

ADB9045I   It should be there.

Explanation: DB2 Admin is attempting to replace an existing version file, but the version file being replaced does not exist.

System action: Processing continues.

User response: Verify that having the version file replaced is the action that you want. Ensure that the version qualifier and version name are correct if you want the version replaced. If you do want to replace an existing version file, change the action to ADD instead of REPLACE in the input parameters to the DB2 Admin program.

ADB9046E   The specified version *version_identifier* **was not found in the database.**

Explanation: DB2 is attempting to replace an existing version file with a version file that is being imported, but the version file being replaced does not exist.

System action: A return code of 12 is set, and processing stops.

User response: Verify that having the version file replaced is the action that you want. Ensure that the version qualifier and version name are correct if you want the version replaced. If you do want to replace an existing version file, change the action to ADD instead of REPLACE in the input parameters to the DB2 Admin program.

ADB9047I   The version ID is *version_identifier*.

Explanation: A version with the identified version ID is being processed.

System action: Processing continues.

User response: None.

ADB9048I   The specified version *version_qualifier.version_name* **was not found in the database.**

Explanation: The version that is being processed should replace an existing version, but that version does not exist.

System action: A return code of 12 is set, and processing stops.

User response: Correct the version qualifier, version name, or both to identify a valid version, and resubmit the request.

ADB9049I   Scope object records are being processed.

Explanation: The process to extract version scope object definitions has started.

System action: Processing continues.

User response: None.

ADB9050I   Version *version_qualifier.version_name* **is being extracted.**

Explanation: A version is needed to process the request, and the identified version is being extracted.

System action: Processing continues.

User response: None.

ADB9051E   The version name, qualifier, or both for version ID *version_identifier* **is null in the database.**

Explanation: DB2 Admin is trying to replace a delta version file, but a delta version file is not found for the version identifier that is provided as input to the DB2 Admin program.

System action: A return code of 12 is set, and processing stops.

User response: Report this internal error to IBM Software Support.

ADB9052W   No scope object records for scope ID *scope_identifier* **were found in the database.**

Explanation: The version that was created might be incomplete because there were no objects defined for the scope that was specified for the version.

System action: A return code of 4 is set, and processing continues.

User response: Complete the definition of the scope by editing the scope and adding objects to it.

ADB9057W   A version already exists with the specified version name.

Explanation: Auto mode is in effect, so the base version will be created with a name like AUTO: and CURTS.

User response: None.

ADB9059W   The version level *version_level version_name* **having a type of** *version_type* **is not supported.**

Explanation: An unknown version level for the specified version record was found in the database.

User response: Verify that the unknown base version record is valid. Return code of 4 is set and processing continues.

ADB9060I   The processing for the ignore or mask begins.

Explanation:

System action: Processing continues.

User response: None.

ADB9061E   An error occurred while processing the ignore or mask request.

Explanation: The DB2 Admin program that processes ignores and masks has encountered an error.

System action: A return code of 8 is set, and processing stops.

User response: See the previously issued message, which provides details about the error.

ADB9062I   The processing for the ignore or mask completed successfully.

Explanation: This report message indicates that DB2 Admin has completed the processing for the ignore or mask successfully.

System action: None.

User response: None.

ADB9063E   The input parameter *input_keyword* for the ignore or mask request was not provided.

Explanation: The ignore or mask could not be processed because the input information that the DB2 Admin program needs was not provided. This is an internal error.

System action: A return code of 12 is set, and processing stops.

User response: Report this internal error to IBM Software Support.

ADB9064E   *op_parameter_value* is not a valid value for the OP parameter.

Explanation: The ignore or mask could not be processed because the input to the DB2 Admin program that processes ignores and masks did not contain a valid value for the OP parameter. This is an internal error.

System action: A return code of 12 is set, and processing stops.

User response: Report this internal error to IBM Software Support.

ADB9065E   *type_parameter_value* is not a valid value for the Type parameter.

Explanation: The ignore or mask could not be processed because the input to the DB2 Admin program that processes ignores and masks did not contain a valid value for the Type parameter. This is an internal error.

System action: A return code of 12 is set, and processing stops.

User response: Report this internal error to IBM Software Support.

ADB9067I   Ignore *ignore_owner.ignore_name* was inserted to database.

Explanation: The request to add an ignore in the Change Management database was successful.

System action: None.

User response: None.

ADB9068E   The definition of ignore *ignore_owner.ignore_name* is incomplete (no ignore lines exist).

Explanation: The ignore cannot be used because its definition is empty.

System action: A return code of 12 is set, and processing stops.

User response: Complete the definition of the ignore by editing the ignore and specifying ignore fields, and resubmit the request.

ADB9069E   Ignore *ignore_owner.ignore_name* does not exist.

Explanation: The request required the use of an ignore and could not be processed because the specified ignore does not exist.

System action: A return code of 12 is set, and processing stops.

User response: Ensure that the correct ignore owner, scope name, or both was specified. Or, create an ignore with the owner and name that was specified. Then, resubmit the request.

ADB9070I   Ignore *ignore_owner.ignore_name* was retrieved from database.

Explanation: The request was processed because the required ignore exists.

System action: None.

User response: None.

ADB9071I   **Mask** *mask_owner.mask_name* **was inserted to database.**

Explanation: The request to add a mask in the Change Management database was successful.

System action: None.

User response: None.

ADB9072E   **Mask** *mask_owner.mask_name* **does not exist.**

Explanation: The request required the use of a mask and could not be processed because the specified mask does not exist.

System action: A return code of 12 is set, and processing stops.

User response: Ensure that the correct mask owner or mask name was specified. Or, create a mask with the specified owner and name and resubmit the request.

ADB9073W   **The definition of mask** *mask_owner.mask_name* **is incomplete (no mask lines exist).**

Explanation: The mask cannot be used because its definition is empty.

System action: A return code of 4 is set, processing continues, and no system action taken.

User response: If you do not intend to use the empty mask, complete the definition of the mask by editing the mask and specifying mask lines. Then, resubmit the request.

ADB9074IE   **Mask** *mask_owner.mask_name* **was retrieved from the database.**

Explanation: The request was processed because the required mask exists.

System action: None.

User response: None.

ADB9075I   **The processing for an ignore or mask is ending.**

Explanation: This report message indicates that DB2 Admin has finished processing an ignore or a mask.

System action: None.

User response: None.

ADB9076E   **The DD statement for** *dd_name* **is missing.**

Explanation: The JCL for the job is missing the identified DD statement.

System action: A return code of 12 is set, and processing stops.

User response: Supply the missing DD statement, and resubmit the job.

ADB9078E   **The specified base version** *owner*, *name* **has an unsupported version level:** *version_level*.

Explanation: The specified base version cannot be used because it contains an earlier version level than the currently supported version. The version level of the base version is located in the CM ADBCVERSION table, TYPE='B'.

System action: The error message is displayed. Return to the previous panel to restart the process.

User response: Create the CM version again using the current release. Admin tool will re-create a new version level.

ADB9110I   **The status of the following changes will be set to DEFINED:**

Explanation: When a recover change is being run, any pending changes to the objects within the recover change are set to DEFINED status. The original change of the recover change is also set to DEFINED status. The original change supersedes any pending changes for the objects within the original change. The pending changes that were superseded are set to DEFINED status. This message introduces the list of the changes that are set to DEFINED status. Message ADB9113 is issued after this message to list each change that is set to DEFINED status.

System action: Processing continues.

User response: Review the list of changes that is displayed after this message to understand which changes are set to DEFINED status when the change is recovered.

ADB9111I   **Owner.Name**

Explanation: Messages ADB9110, ADB9111, ADB9112, and ADB9113 are issued in conjunction with each other. This message provides a heading to identify the owner and the name of the changes that are listed by message ADB9113.

System action: Processing continues.

User response: See message ADB9110.

ADB9112I   ----–--------------------------------------

Explanation: Messages ADB9110, ADB9111, ADB9112, and ADB9113 are issued in conjunction with each other. This message provides a heading for message ADB9113.

System action: Processing continues.

User response: See message ADB9110.

ADB9113I *change_owner.change_name*

Explanation: Messages ADB9110, ADB9111, ADB9112, and ADB9113 are issued in conjunction with each other. This message lists the owner and name of each change that is set to DEFINED status when you recover the change.

System action: Processing continues.

User response: See message ADB9110.

ADB9300E Change *change_owner.change_name* **cannot be recovered until the following changes are recovered in the order that they are specified. The list contains those changes that completed after the change to recover completed and have not been recovered. They modify the same or related objects as those in the change to recover and, hence, the recover change itself. Rcvr Order Owner.Name -----**

Explanation: An attempt is being made to recover a change that cannot be recovered because other changes must be recovered first. The accompanying messages provide a list of the changes that must be recovered first.

System action: Processing stops.

User response: Recover the list of changes in the order that is specified.

ADB9304E **This change cannot be recovered because it does not have a recover change or its recover change is not in the ANALYZED state.**

Explanation: An attempt is being made to recover a change that cannot be recovered because it does not have a recover change or its recover change is not in ANALYZED status.

System action: Processing stops.

User response: Ensure that each change currently being recovered that is, the change is in RUNNING status) completes. Otherwise, create a new change to undo the changes made by this change.

ADB9305I **The following pending changes will be set to DEFINED status. These changes modify the same or related objects as those in the change to recover and, hence, the recover change itself. Owner.Name**

Explanation: A change is being recovered, and there are pending changes for the objects that are affected by the change to recover. The pending changes will be set to DEFINED status. The accompanying messages provide a list the changes that will be set to DEFINED status.

System action: Processing continues.

User response: None.

ADB9306I **This change can be recovered. No other changes that modify the same or related objects completed after the change completed, and there are no pending changes that modify the same or related objects.**

Explanation: A change is being recovered, and this informational message indicates that there are no other changes that need to be recovered first and that there are no pending changes for the affected objects.

System action: Processing continues.

User response: None.

ADB9307E **This change cannot be recovered because the WSL and JCL files for the recover change do not exist.**

Explanation: An attempt was made to recover a change, and the WSL and JCL files that are required to recover the change do not exist. The change cannot be recovered.

System action: Processing stops.

User response: Create a new change to undo the changes made by this change.

ADB9308E **The JCL file for the recover change does not exist. An error occurred while a temporary JCL file for the recover WSL was being created.**

Explanation: An attempt was made to recover a change, and the JCL file for the recover job that is required to recover the change does not exist.

System action: Processing stops.

User response: Create a new change to undo the changes made by this change.

ADB9351E **An error occurred when the change status was updated. Neither the old or new change status values match the current change status:** *current_change_status***.**

Explanation: The request to update the change status was invalid.

System action: Processing stops.

User response: If you submitted a run job, ensure that you analyze the change before running it. If you submitted an analyze job, ensure that the change is in DEFINED or ANALYZED status before submitting the analyze job.

ADB9352E **The specified change** *change_ID* **does not exist.**

Explanation: A request was made to update the change status for a change ID that does not exist.

System action: No system action is taken.

User response: Try generating a new run job or re-analyze the change.

ADB9353E **SQL error** *SQL_error_code* **occurred while the Change Management database was being accessed.**

Explanation: An unexpected SQL error occurred while accessing DB2.

System action: None.

User response: Fix the problem and try again.

ADB9400I **The change was registered successfully. Changeid:** *Change_ID*

Explanation: The specified change was successfully registered.

System action: No action is required.

User response: Processing continues.

ADB9401E **Registration has failed. Error in input parameters: Change Owner:** *Change_Owner* **Change Name:** *Change_Name* **Change Type:** *Change_Type*

Explanation: There was an error in one of the input parameters and the registration has failed.

System action: Processing stops.

User response: Correct the parameters and try again.

ADB9403E **Registration has failed. Error in input parameters: Start Mode:** *Start_Mode* **Register Mode:** *Register_Mode* **Input Type:** *Input_Type* **Input Name:** *Input_Name*

Explanation: An error in one or more of the input parameters has caused the registration to fail.

System action: Processing stops.

User response: Correct the parameters and try again.

ADB9405E **Error registering the change. Another change already exists with: Change Owner:** *Change_Owner* **Change Name:** *Change_Name* **Change Type:** *Change_Type*

Explanation: The change cannot be registered because the change owner, name or type already exists.

System action: Processing stops.

User response: Modify the change owner and/or name and try again.

ADB9406E **Change does not exist. Change Owner:** *Change_Owner* **Change Name:** *Change_Name* **Change Type:** *Change_Type*

Explanation: The change must exist for including into an existing change.

System action: Processing stops.

User response: Ensure that the change already exists.

ADB9407E **ChangeID for the original change must be provided to recover. ChangeID:** *Change_ID*

Explanation:

System action: Processing stops.

User response: Provide the changeid for the orginal change and try again.

ADB9409E **Registration could not be completed. Reason Code:** *Reason_Code* **Reason:** *Reason* **Change ID:** *Change_ID*

Explanation: The registration could not be completed for the specified reason.

System action: Processing stops.

User response: Correct the error and try again.

ADB9410E **The restart failed. A change ID is required to restart a change.**

Explanation: You must specify the change ID of the change to restart.

System action: Processing stops.

User response: Specify the change ID of the change to restart.

ADB9411E **The change is not in restartable status. Change Status:** *Change_Status*

Explanation: Changes in INITIAL, DEFINED or ANALYZED status are eligible for restart.

System action: Processing stops.

User response: Ensure that the change is in restartable status.

ADB9412E **Too few parameters were specified to associate a target. Target Name:** *Target_Name*

Explanation: You must specify the correct number of parameters for the specified target.

System action: Processing stops.

User response: Specify the missing parameters and try again.

ADB9413E **The specified target is already associated with the MT Change. Target Name:** *Target_Name* **Target Change Owner:** *Target_Change_Owner* **Target Change Name:** *Target_Change_Name* **Target Change Status:** *Target_Change_Status*

Explanation: The specified target is already associated with the multi-target change.

System action: Processing stops.

User response: Specify a different target profile and try again.

ADB9414E **The target profile was not found. Target Profile Name:** *Target_Profile_Name*

Explanation: The specified target profile name was not found

System action: Processing stops.

User response: Specify an existing target profile and try again.

ADB9418E **A multi-target change is already registered that uses either the same mask or no mask was specified. Details of the existing change: Change ID** *change_ID*, **Change Owner:** *change_owner*, **Change Name:** *change_name*, **Change Status:** *change_status*.

Explanation: You cannot use the same mask multiple times because it might result in redundant changes to objects.

System action: Processing continues with the next change.

User response: Specify a different mask and try the operation again.

ADB9419I **An existing target change was restarted. Change ID:** *change_ID*.

Explanation: A request to register a multi-target change has been received; however, a change with the same mask already exists in INITIAL status. An attempt was made to restart the existing change rather than registering it as a duplicate change. The success or failure of restarting the change is reported.

System action: Processing continues.

User response: None required.

ADB9421E **Cannot replace a change with Change Type:** *Change_Type*.

Explanation: A request to replace a change was received but cannot be processed. Only changes with the change type 'CHANGE' can be replaced.

System action: Processing stops.

User response: Modify the change owner or change name to select another existing change that has the change type 'CHANGE', or to create a new change and then try again.

ADB9422E **Cannot replace a change with status:** *Change_Status*.

Explanation: A request to replace a change was received but cannot be processed because of an existing change. The existing change must have a change status of initial, defined, analyzed, or canceled in order to be replaced.

System action: Processing stops.

User response: Modify the change owner or change name to select another existing change that has the change type 'CHANGE', or to create a new change and then try again.

ADB9424E **Registration failed to replace the change. Change ID . . . . :** *change_ID*, **Change Owner . . . . :** *change_owner*, **Change Name . . . . :** *change_name*

Explanation: Error occurred replacing a change. Review other messages in the report to ascertain the failure.

System action: Processing stops.

User response: Look for other messages that can help identify the reason that the replace change request failed. Correct the error and try again.

ADB9426E   **Check the Work Load Manager (WLM) environment started task** *wlm_environment_name* **for additional messages and check the WLM settings.**

Explanation: The call to the ADBCRSP procedure (the multiple target change stored procedure) failed.

System action: Processing terminates abnormally.

User response: Check the task started in the Workload Manager (WLM) environment *wlm_environment_name* for additional messages. Also, check with the User's Guide to confirm that the WLM settings are correct.

ADB9735E   **You requested that the DDL be generated from a base version, but the version does not exist**

Explanation: If the type is USER, the owner and name values are the base version owner and name that you specified. Otherwise, the type indicates the type of base version you requested along with the change owner and name values that you specified.

System action: Processing ends.

User response: Ensure that the specified base version type exists for the specified change. If the DDL from a user-specified base version was requested, ensure that the version exists.

ADB9736E   **You requested that the DDL be generated from a base version, but the version requested is not a base version.**

Explanation: If the type is USER, the owner and name values are the base version owner and name that you specified. Otherwise, the type indicates the type of base version you requested, along with the change owner and name values that you specified.

System action: Processing ends.

User response: If the DDL from a user-specified base version was requested, ensure that the version owner and name you specified matches an existing base version and not a delta version. If you did not request the DDL from a user-specified base version, you should report this to IBM.

ADB9751E   **The multi-target change cannot be analyzed because at least one action parameter other than action_analyze_change is set to Y.**

Explanation: Analyzing a multi-target change will pick up all the corresponding target changes on the local DB2 subsystem for analysis. However, you cannot combine this action with other actions like run or recover.

System action: Processing stops.

User response: Fix the parameter value and try again.

ADB9752E   **The change management batch parameter option** *<option>* **is not supported for the multi-target change analyze or run process.**

System action: Processing stops.

User response: Remove the parameter value and try again.

ADB9908I   **ADB9908I Processing change: Owner . . :** *owner_name*, **Name : . . .***name*

Explanation: Data for the identified change is being moved from the local backup tables into the identified InfoSphere Optim Configuration Manager repository database.

System action: No system action is taken.

User response: No action to take.

ADB9909I   **Statement information: Approximate run timestamp . . :***timestamp*, **Statement type** *statement_type* **Object type** *object_type,* **, Object qualifier** *object_qualifier*, **object name** *object_name*

Explanation: Data for the identified statement is being moved from the local backup tables into the identified InfoSphere Optim Configuration Manager repository database.

System action: No system action is taken.

User response: No action to take.

ADB9910E   **The change information could not be stored into the InfoSphere Optim Configuration Manager repository database. A severe error occurred.**

Explanation: The InfoSphere Optim Configuration Manager repository database is not available. Look for other SQL error messages for the details of the error.

System action: Processing stops.

User response: Correct the error and try again.

ADB9911E   **The change information could not be stored into the InfoSphere Optim Configuration Manager repository database. The action on error setting is** *action_on_error*. **A ROLLBACK will be done and processing will stop.**

Explanation: The InfoSphere Optim Configuration Manager repository database is not available. Look for other SQL error messages for the details of the error.

System action: Processing stops.

User response: Correct the error and try again.

ADB9912W **The change information could not be stored into the InfoSphere Optim Configuration Manager repository database. The action on error setting** *action_on_error*. **The data was stored into backup tables on the local system. Processing continues.**

Explanation: The InfoSphere Optim Configuration Manager repository database is not available. Look for other SQL error messages for the details of the error. The data was instead stored in backup tables on the local system.

System action: Processing continues.

User response: When the InfoSphere Optim Configuration Manager repository database is available, the DB2 Admin ADBLIM program can be run to move data from the backup tables on the local system to the InfoSphere Optim Configuration Manager repository database.

ADB9913E **The change information could not be stored into the backup tables on the local system. A severe error occurred.**

Explanation: A severe error occurred while attempting to write to the backup tables on the local system. Look for other SQL error messages for the details of the error.

System action: Processing stops.

User response: Correct the error and try again.

ADB9914E **The change information could not be stored into the InfoSphere Optim Configuration Manager repository database, or in the backup tables on the local system. The action on error setting is** *action_on_error*. **Processing stops.**

Explanation: The InfoSphere Optim Configuration Manager repository database is not available, and the backup tables on the local system are not available. Look for other SQL error messages for the details of the error.

System action: Processing stops.

User response: When the InfoSphere Optim Configuration Manager repository database is available, the DB2 Admin ADBLIM program can be run to move data from the backup tables on the local system to the InfoSphere Optim Configuration Manager repository database.

ADB9915E **The change information could not be stored into the InfoSphere Optim Configuration Manager repository database, or in the backup tables on the local system. The action on error setting is OVERRIDE. Processing stops. You can specify to override the error and continue processing the change. If the OCM repository database and the backup tables on the local system are not available, DB2 Admin will continue processing the change but information about the change will not be recorded. To override the error in batch: - When using CM batch, specify the OVR_CONFIGDB_ERROR = 'Y' parameter in the PARMS DD file. When not using CM batch, specify the OVR_CONFIGDB_ERROR = 'Y' parameter in the ADBTEPIN DD file. To override the error online, specify YES to the override option.**

Explanation: The InfoSphere Optim Configuration Manager repository database is not available, and the backup tables on the local system are not available. Look for other SQL error messages for the details of the error.

System action: Processing stops.

User response: When the InfoSphere Optim Configuration Manager repository database is available, the DB2 Admin ADBLIM program can be run to move data from the backup tables on the local system to the InfoSphere Optim Configuration Manager repository database.

ADB9916W **The change information could not be stored into the InfoSphere Optim Configuration Manager repository database, or in the backup tables on the local system. The action on error setting is** *action_on_error*. **The OVR_CONFIGDB_ERROR parameter was set to 'YES', so the information about the changes made will not be stored in the InfoSphere Optim Configuration Manager repository database, or the local backup tables.**

Explanation: The InfoSphere Optim Configuration Manager repository database is not available, and the backup tables on the local system are not available. Look for other SQL error messages for the details of the error.

System action: Processing continues.

User response: Once the InfoSphere Optim Configuration Manager repository database is available, the DB2 Admin ADBLIM program can be run to move data from the backup tables on the local system to the

InfoSphere Optim Configuration Manager repository database.

ADB9918W   For SHRLEVEL CHANGE processing, RECLUSTER NO is always enforced by the REORG TABLESPACE utility.

System action: Processing continues.

User response: No action is required.

ADBA016W   For SHRLEVEL CHANGE processing, RECLUSTER NO is always enforced.

Explanation: This warning message indicates that RECLUSTER NO is always enforced for SHRLEVEL CHANGE processing.

System action: Processing continues.

User response: No action is required.

ADBC099E   There is a WSL mismatch. The WSLs did not compare equally.

Explanation: The run-time WSL and the analyze-time WSL are different.

System action: Processing stops.

User response: Examine the environment to determine whether the change needs to be re-analyzed.

ADBC007E   Invalid field name in the IGNORES file record.

Explanation: The IGNORES file contains invalid ignore field specifications which can not be processed.

System action: Processing stops.

User response: Review the ignore field specifications in the IGNORES file and make sure all the fields specified are listed as supported catalog table ignore fields (refer to the IBM DB2 Administration Tool for z/OS User's Guide), or redefine the ignores by specifying the ignore fields on the Specify Ignore Fields panel.

ADBC016E   The object *object_owner*.*object_name* **exists.**

Explanation: An object *object_owner*.*object_name* already exist. Specify a new owner and name.

System action: Processing stops.

User response: Specify an owner and name so that the combination of owner and name is unique from objects that already exist.

ADBC027E   Target profile not found

Explanation: The specified target profile, `target_profile` was not found.

System action: Processing ends.

User response: Specify a valid target profile and try the operation again.

ADBC030E   Register Failed.

Explanation: The reason code and reason for failure are displayed as part of the long message. If the failure occurred during a call to a register interface, the return code from the interface is displayed as the reason code.

System action: The process of registering a change terminated.

User response: If the reason information does not help to resolve the issue, contact IBM® support to report the message.

ADBC060E   Pending changes exist that have an incompatible record layout.

Explanation: One or more pending changes exist that have an internal record layout that is incompatible with the current version of the product.

System action: Processing is halted to prevent use of the incompatible records.

User response: You can use the RST line command from the change management dialog (panel ADB2C11) to reset each of the incompatible record layouts.

ADBC066W   No target changes to process.

Explanation: An attempt was made to export multi-target information into a dataset on the target system in an effort to communicate target updates to the central system.

System action: Processing stops. No information is written to the TGTINFO file.

User response: Ensure that a list of multi-target changes is provided.

ADBC068E   The specified base version *owner*, *name* **has an unsupported version level:** *version_level*.

Explanation: The version level is not supported.

System action: Processing stops.

User response: Specify the appropriate version level and try again.

ADBC081   **The JCL data set or data set member does not exist.**

Explanation: This message is issued when the ER line command was issued to edit the run job or promote job or the EA line command was issued to edit the analyze job for a change, and the JCL data set or data set member does not exist.

System action: Processing stops

User response: If the ER line command was issued, the action to take depends on the type of change and its status:

- For a change type of CHANGE:
  - If the status is ANALYZED, issue RN line command to rebuild the run job.
  - If the status is RUNNING, use the job that is stored in SDSF.
  - If the status is not ANALYZED or RUNNING, the run job is no longer needed and no action is needed.
- For a change type of COMPARE:
  - If the status is DEFINED, the promote job is no longer valid. Create the promote job again.
  - If the status is COMPLETE, the promote job is no longer needed because the job has already been executed successfully and no action is needed.

If the EA line command was issued, the action to take depends on the status of the change:

- If the status is ANALYZED, RUNNING, or COMPLETE, the analyze job that was used to analyze the change is not accessible. No action is needed.
- If the status is DEFINED, re-analyze the change.
- If the status is none of the above, get the change into DEFINED status and then re-analyze the change.

ADBC082   **The change cannot be run because there are prerequisite changes that must be run first.**

Explanation: The RN line command was issued to build a run job, but the change has prerequisite changes that must be run first.

System action: Processing stops.

User response: Run the prerequisite changes before re-issuing the RN line command to build the run job for the change. You can issue the PQ line command on the Changes panel (ADB2C11) to get a list of the prerequisite changes.

ADBC083E   **A RESTART parameter was not automatically added because the job card is missing. The job needs to be restarted.**

Explanation: The ER line command was issued so that the job to run (or promote) a change in RUNNING status could be edited and then resubmitted. DB2 Admin was unable to automatically add the RESTART parameter to have the job restarted at the identified step because the job card is missing.

System action: Processing continues, and the JCL to run the job is displayed in edit mode.

User response: Add a job card to the JCL that includes a RESTART parameter so that the job is restarted at the identified step. Then, submit the job.

ADBC084E   **A RESTART parameter was not automatically added to restart the step that runs program ADBTEP2 because the step could not be found.**

Explanation: The ER line command was issued so that the job to run (or promote) a change in RUNNING status could be edited and then resubmitted. DB2 Admin was unable to automatically add the RESTART parameter to have the job restarted at the step that runs program ADBTEP2 because DB2 Admin could not find the step that runs that program.

System action: Processing continues, and the JCL to run the job is displayed in edit mode.

User response: Ensure that the JCL is valid. Then, submit the job.

ADBC085E   **The RESTART parameter was not automatically added to the job card because either the step that runs the program could not be found or the job card is missing.**

Explanation: The ER line command was issued so that the job to run (or promote) a change in RUNNING status could be edited and then resubmitted. DB2 Admin was unable to automatically add the RESTART parameter. Either the step that runs the identified program could not be found or the job card is missing.

System action: Processing continues, and the JCL to run the job is displayed in edit mode.

User response: Ensure that the JCL is valid. Then, submit the job.

ADBC100E   **The *owner*.*name* change does not exist.**

Explanation: An attempt was made to delete a change that does not exist.

System action: Processing ends.

User response: Refresh the panel to retrieve the current list of changes.

ADBC101E   You do not have the privilege to delete the *owner.name* **change according to the definition of the delete change view (ADBCHGV1).**

Explanation: The delete change view (ADBCHGV1) has been defined in a way that prevents you from deleting the change.

System action: Processing ends.

User response: Check with the system administrator who installed DB2 Admin and enabled Change Management.

ADBC102E   The *owner.name* **change cannot be deleted because the change does not satisfy the delete criteria.**

Explanation: The change cannot be deleted because the change does not meet the criteria for being dropped. To be dropped, a change must meet one of the following criteria:
- The status of the change is CANCELED
- The status of the change is FAILED and the type is FAST
- The type of the change is COMPARE

System action: Processing ends.

User response: Put the change into a status such that the criteria to delete a change is met, and then try the DEL line command to delete the change again.

ADBC103E   You do not have the privilege to run the delete change command.

Explanation: You have not been given the privilege to delete changes. This error usually means that an SQLCODE -922 was received while an attempt was made to run the ADBCDCH plan.

System action: Processing ends.

User response: Check with the system administrator who sets up the DB2 Admin plans and packages to request access to the ADBCDCH plan.

ADBC104E   The delete change command is not enabled.

Explanation: DB2 Admin has not been configured to enable the delete change command. This error usually means that an SQLCODE -805 was received while an attempt was made to run the ADBCDCH package.

System action: Processing ends.

User response: Check with the system administrator who sets up the DB2 Admin plans and packages to

request the appropriate set up of the ADBCDCH package and plan.

ADBC154E   Incompatible data sets. A data set with multi-target change content and a data set without multi-target change content cannot be imported together.

Explanation: A data set with multi-target change content cannot be imported with other data set(s). A data set with multi-target change content must be imported alone.

System action: Processing ends.

User response: Perform the import using a single multi-target change content data set, and another import using all non-multi-target change data sets.

ADBC155E   Incompatible data sets. Multiple data sets with multi-target change content cannot be imported at the same time.

Explanation: A data set with multi-target change content cannot be imported at the same time as other data sets with multi-target change content.

System action: Processing ends.

User response: Import data sets with multi-target change content one at a time.

ADBC301E   The EDIT line command requires installation and enablement of the DB2 Table Editor.

Explanation: The DB2 Table Editor is not installed and enabled.

System action: Processing ends.

User response: Check with the Tools Customizer administrator for the DB2 Admin for assistance.

ADBC302E   The EDIT line command is not enabled.

Explanation: The EDIT line command is not enabled.

System action: Processing ends.

User response: Check with the Tools Customizer administrator, and ensure that the `Enable DB2 Table Editor` parameter is set to YES.

ADBC303E   The EDIT line command cannot locate the DB2 Table Editor library.

Explanation: The EDIT line command cannot locate the DB2 Table Editor library.

System action: Processing ends.

User response: Check with the Tools Customizer administrator, and ensure that the DB2 Table Editor CLIST library has been specified.

ADBC313E   Mask value error

Explanation: The mask value in the To column cannot start with a comma.

System action: Processing stops.

User response: Remove the comma.

ADBC314E   Mask value error

Explanation: The mask value in the From column cannot end with a comma.

System action: Processing stops.

User response: Remove the comma.

ADBG001E   Verification of the data set failed. The input data set must be either fixed length (F/FB) with LRECL=80 or variable length (V/VB) with LRECL between 16000 and 16384.

Explanation: The input dataset must be either fixed length with a record length of 80 bytes or variable length with record length between 16000 and 16384 bytes.

System action: Processing stops.

User response: Specify a valid record format (RECFM) and record length (LRECL) for the data set.

ADBG002E   Verify failed -- Value for DSORG is not supported.

Explanation: The data set is a type that cannot be processed.

System action: Processing stops.

User response: Specify a member name and try again.

ADBG004E   No member name specified

Explanation: A member name is required for the PDS or LIBRARY.

System action: Processing ends.

User response: Add a member name and try the operation again.

ADBG009E   Invalid entry specified

Explanation: Either an invalid directory block number was specified for the data set name type or an invalid data set name type was specified for the directory block number.

System action: Processing ends.

User response: Modify the directory blocks value and try the operation again.

ADBG010E   Verification has failed

Explanation: The value specified for LRECL, RECFM, or DSNTYPE does not match the value for the existing data set.

System action: Processing ends

User response: Modify the value for the parameter and try the operation again.

ADBG011E   Data set does not exist

Explanation: The specified data set or member does not exist.

System action: Processing continues.

User response: Ensure that the specified data set exists and try the operation again.

ADBG013E   All columns have been deleted except for one or more hidden columns. A table cannot contain only hidden columns.

Explanation: The table from which you are deleting columns contains hidden columns. In DB2, a table cannot contain only hidden columns.

System action: None.

User response: No action required.

ADBM001E   Too many columns

Explanation: The maximum number of ORDER BY columns that can be defined is 10.

User response: Reduce the number of columns that have been selected, and try again.

ADBM002E   Invalid column

Explanation: The column with the name COLnnnn can not be used in an ORDER BY clause in DB2 Admin because the column is the result of an expression.

User response: Remove the column from the list of columns that are designated to be saved in the ORDER BY clause.

ADBM003E   ORDER command not valid

Explanation: The ORDER command cannot be used on this panel because DB2 Admin requires that the rows be in a defined sequence.

User response: Use valid commands to configure the current panel. Valid commands are listed on the panel.

ADBM005E   Save failed

Explanation: The ORDER BY clause was not saved. Examine the ISPF log data set.

User response: See the error that was written in the ISPF log data set. Resolve the problem and retry.

ADBM006E   ORDER BY error

Explanation: The ORDER BY clause for the panel caused SQLCODE -208 and the column in error was removed from the SELECT statement. Remove the column from the ORDER BY clause by using the ORDER command.

User response: Exit this panel and return to the previous panel to remove the column, and try again.

ADBM009E   Promote failed

Explanation: The promotion of the ORDER BY clause to the installation default data set failed. Examine the ISPF log data set.

User response: See the error that was written in the ISPF log data set. Resolve the problem and retry.

ADBM024E   The overwrite value that is specified for the SEGSIZE must be an integer that is a multiple of 4.

Explanation: The mask contains a value for SEGSIZE that is not valid.

System action: A return code of 1012 is set, and processing stops.

ADBM025E   The overwrite value that is specified for COMPRESS must be YES or NO.

Explanation: The mask contains a value for COMPRESS that is not valid.

System action: A return code of 1012 is set, and processing stops.

User response: Change the mask definition to specify a value for SEGSIZE that is a multiple of 4, and then resubmit the job.

ADBM026E   The overwrite value for DSSIZE must be a numeric value that is followed by the character 'G'.

Explanation: The use of masking was specified, and the value that is specified for DSSIZE is not valid.

System action: A return code of 1012 is set, and processing stops.

User response: Correct the definition of the mask. If a specific value is specified for DSSIZE, ensure that the value is an integer value that is followed by the

character 'G', for example, 8G. If a REXX user exit is specified for DSSIZE, ensure that the REXX user exit is coded so that it returns an integer value followed with character 'G'. After the corrections are made, resubmit the job.

ADBM027E   The overwrite value for *space_allocation_quantity_attribute* **must be a numeric value.**

Explanation: The use of masking was specified, and the value that is specified for *space_allocation_quantity_attribute* (PRIQTY, TSPRIQTY, IXPRIQTY) is not valid.

System action: A return code of 1012 is set, and processing stops.

User response: Correct the definition of the mask. If a specific value is specified for *space_allocation_quantity_attribute*, ensure that the value is an integer value. If a REXX user exit is specified for *space_allocation_quantity_attribute*, ensure that the REXX user exit is coded so that it returns an integer value. After the corrections are made, resubmit the job.

ADBM028E   The overwrite value for *space_allocation_quantity_attribute* **must be a numeric value.**

Explanation: The use of masking was specified, and the value that is specified for *space_allocation_quantity_attribute* (SECQTY, TSSECQTY, or IXSECQTY)) is not valid.

System action: A return code of 1012 is set, and processing stops.

User response: Correct the definition of the mask. If a specific value is specified for *space_allocation_quantity_attribute*, ensure that the value is an integer value. If a REXX user exit is specified for *space_allocation_quantity_attribute*, ensure that the REXX user exit is coded so that it returns an integer value. After the corrections are made, resubmit the job.

ADBM029E   The overwrite value for DEFER must be YES or NO.

Explanation: The use of masking was specified, and the value that is specified for DEFER is not valid.

System action: A return code of 1012 is set, and processing stops.

User response: Correct the definition of the mask. If a specific value is specified for DEFER, ensure that the value is YES or NO. If a REXX user exit is specified for DEFER, ensure that the REXX user exit is coded so that it returns the value YES or NO. After the corrections are made, resubmit the job.

ADBM030E   The overwrite value for *define_attribute* **must be YES or NO.**

Explanation: The use of masking was specified, and the value that is specified for *define_attribute* (DEFINE, TSDEFINE, or IXDEFINE) is not valid.

System action: A return code of 1012 is set, and processing stops.

User response: Correct the definition of the mask. If a specific value is specified for *define_attribute*, ensure that the value is YES or NO. If a REXX user exit is specified for *define_attribute*, ensure that the REXX user exit is coded so that it returns the value YES or NO. After the corrections are made, resubmit the job.

ADBM209E   The column is a pending drop column so it cannot be dropped.

Explanation: The DROP line command was entered on the ADB21TC panel but the column is already marked to be dropped and the table space is in advisory REORG-pending status.

System action: Processing stops.

User response: A column marked as pending drop cannot be dropped. Select a different valid column.

ADBM703E   The selected dialog name does not exist.

Explanation: The selected dialog name might have been renamed or deleted by another user.

System action: Processing stops.

User response: Enter REFRESH on the command line, and then select a dialog name that is available.

ADBM706E   The *&zcmd* **command cannot be used with the line command that you specified. Remove the** *&zcmd* **command and then proceed.**

Explanation: The command cannot be used with the line command.

System action: Processing stops.

User response: Remove the command and press Enter. The product will continue to execute the line commands one by one.

ADBM708E   A template syntax error occurred while building the apply job or work statement list.

Explanation: DB2 Admin detected a syntax error for the user-defined template while building the apply job or work statement list.

System action: No system action is taken.

User response: Modify the definition for the user-defined template.

ADBU000E   The UNLOAD utility does not support LOB table spaces.

Explanation: The DB2 UNLOAD utility will not process a LOB table space.

System action: Processing stops.

User response: Perform the unload on the base table space. The unload will contain the data from the LOB table space.

ADBU012E   For a partitioned table space, the Repair Utility with LEVELID option must be initiated at the partition level. Enter S in the line command field. Subsequently, enter SP in the line command field, then enter the utility dialog for the specific table space partition.

Explanation: The REPAIR LEVELID utility cannot operate at the table space level. It must be initiated at the partition level.

System action: The system waits.

User response: Press F3 to return to the VIEW panel, then enter S by the view name. On the subsequent panel, entert SP for the table space that is shown. On the subsequent panel, enter the UTIL line command for the specific table space partition.

ADBZ001E   Table cannot be archived because *message*

Explanation: The table cannot be archived because *message*, where *message* is one of the following:
- no partitions were selected.
- no SYSACCELERATEDTABLES table exists.
- a table is not specified for accelerator.
- XML or LOB columns are present in the table.
- the table is a parent of foreign key relationship.
- the table is not in a partition by range table space.
- the stored procedure ACCEL_ARCHIVE_TABLE does not exist.

System action: Processing ends.

User response: If possible, fix any error conditions and try the operation again.

ADBZ002E   Partition range is invalid because *message*

Explanation: The partition range is invalid because *message*, where *message* is one of the following:
- no spaces are allowed in the range list.
- an invalid character is in the range list.

- of invalid range list syntax.
- the ending part in the range construct must be greater.
- the part specified is greater than the maximum part.

System action: Processing ends.

User response: Change the partition range using valid syntax and partition values and try the operation again.

ADBZ009E   **The attempt to disable or enable incremental updates failed because** *reason*.

Explanation: Disabling or enabling incremental updates failed because *reason*, where *reason* is one of the following:

- stored procedure ACCEL_SET_TABLES_REPLICATION does not exist.
- the associated accelerator is virtual.

System action: Processing ends.

User response: Ensure that the stored procedure exists and that the accelerator is not virtual and try the operation again.

ADBZ104E   **Data set already exists.**

Explanation: The specified sequential data set already exists. You must enter a unique value.

System action: Processing stops.

User response: If you want to reuse the data set, delete the existing data set and a new data set will be generated. If you want to create a new data set, change the modifier so that the data set fully qualified name is different.

ADBZ105E   **Data set already exists.**

Explanation: The specified sequential data set already exists. You must enter a unique value.

System action: Processing stops.

User response: If you want to reuse the data set, delete the existing data set and a new data set will be generated. If you want to create a new data set, change the modifier so that the data set fully qualified name is different.

Tools Customizer troubleshooting

Use this information to diagnose and correct problems that you experience with Tools Customizer.

Gathering diagnostic information

Before you report a problem with Tools Customizer to IBM Software Support, you need to gather the appropriate diagnostic information.

Procedure

Provide the following information for all Tools Customizer problems:
- A clear description of the problem and the steps that are required to re-create the problem
- Relevant screen captures
- All messages that were issued as a result of the problem
- Product release number and the number of the last program temporary fix (PTF) that was installed
- The version of DB2 that you are using and the type and version of the operating system that you are using
- The Tools Customizer trace data set
- The Tools Customizer data store data set and the *high_level_qualifier*.SCCQTENU data set

Determining the trace data set name

You will need to identify the name of the trace data set if you cannot allocate the trace data set, the trace data set runs out of space, or IBM Software Support asks for it.

The name of the trace data set depends on the prefix setting in the TSO profile. To identify the name of the trace data set, you must know the prefix setting.

- If PREFIX is set, the name of the trace data set is *prefix*.CCQ.TRACE, where *prefix* is the TSO prefix that you specified in the profile.
- If NOPREFIX is set, the name of the trace data set is *user_ID*.CCQ.TRACE, where *user_ID* is your TSO user ID.

Tools Customizer messages

Use the information in these messages to help you diagnose and solve Tools Customizer problems.

CCQB000I   The product parameter data was saved in the data store.

Explanation: Changes that were made to the product parameters were saved in the data store.

System action: None.

User response: No action is required.

CCQB001I   The DB2 parameter data was saved in the data store.

Explanation: Changes that were made to the DB2 parameters were saved in the data store.

System action: None.

User response: No action is required.

CCQB002I   The LPAR parameter data was saved in the data store.

Explanation: Changes that were made to the LPAR parameters were saved in the data store.

System action: None.

User response: No action is required.

CCQB003E   At least one step must be selected in a selected task. The selected task is *task_description*.

Explanation: When a task is selected, at least one step must be selected. A selected step is missing from the specified task.

System action: Processing stops.

User response: Select a step in the specified task or deselect the task.

CCQB004I   The required information to run the Discover EXEC was saved in the data store.

Explanation: The data store contains all the information that is required to run the Discover EXEC.

System action: None.

User response: No action is required.

CCQB005E   The conflicting values for the *parameter_name* parameter must be resolved before the information can be saved.

Explanation: Two values for one parameter conflict with each other, and they must be resolved to save the information.

System action: Processing stops.

User response: Resolve the conflicting values for the parameter.

CCQB006E   One row must be selected.

Explanation: One row in the table must be selected.

System action: Processing stops.

User response: Select one row.

CCQB007E   Only one row can be selected.

Explanation: Multiple rows in the table are selected, but only one row is allowed to be selected.

System action: Processing stops.

User response: Select only one row.

CCQC000I   The jobs have been customized on the selected DB2 entries.

Explanation: The jobs were customized on the DB2 entries that were selected.

System action: None.

User response: Press Enter to clear the message.

CCQC001W   The jobs were not generated on one or more of the selected DB2 entries. Press PF3 to check the DB2 entries that were not customized.

Explanation: The product was not customized on one or more of the DB2 entries that were selected.

System action: None.

User response: Press PF3 to see the DB2 entries on which the product was not customized. The status of these DB2 entries is Errors in Customization.

CCQC002I	The edit session was started automatically because values for required parameters are missing or must be verified.

Explanation: If product, LPAR parameters, or DB2 parameters are not defined or if parameter definitions must be verified, an editing session for the undefined or unverified parameters starts automatically.

System action: None.

User response: Define values for all required product, LPAR parameters, or DB2 parameters.

CCQC003W	The *template_name* template in the *library_name* metadata library does not contain any parameters.

Explanation: The specified template does not have parameters.

System action: None.

User response: No action is required.

CCQC004S	The value of the "type" attribute for the *template_name* template in the *library_name* metadata library does not match the value that was previously specified. The value is *value_name*, and the previously specified value is *value_name*.

Explanation: The value of the "type" attribute must match the value that was previously specified.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQC005S	The *template_name* template exceeds the number of allowed templates for a customization sequence. The template is in the *library_name* metadata library.

Explanation: The customization sequence can process only *number* templates. The specified template cannot be processed because the customization sequence already contains the maximum number of templates.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQC006E	The jobs could not be generated for the *group_attach_name* DB2 group attach name.

Explanation: The customization jobs could not be generated for the specified DB2 group attach name.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQC007E	The jobs could not be generated for the *subsystem_ID* DB2 subsystem.

Explanation: The customization jobs could not be generated for the specified DB2 subsystem.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQC008E	The jobs could not be generated for the *member_name* DB2 member.

Explanation: The customization jobs could not be generated for the specified DB2 member.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQC009S	The jobs were not generated for the DB2 entries.

Explanation: One or more errors occurred while customization jobs were being generated for the selected DB2 entries.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQC010S	The *template_name* template could not be accessed in the *library_name* metadata library.

Explanation: The specified template could not be accessed because the user does not have RACF access to the data set, the data set has incorrect data characteristics, or the data set is not cataloged.

System action: Processing stops.

User response: Ensure that you have RACF access to the data set, that the characteristics are correct according to the specifications of the product that you are customizing, and that the data set is cataloged. If

the problem persists, contact IBM Software Support.

CCQC011S   The *template_name* **template could not be written to the** *library_name* **customization library.**

Explanation: The specified template could not be accessed because the user does not have RACF access to the data set, the data set has incorrect data characteristics, or the data set is not cataloged.

System action: Processing stops.

User response: Ensure that you have RACF access to the data set, that the characteristics are correct according to the specifications of the product that you are customizing, and that the data set is cataloged. If the problem persists, contact IBM Software Support.

CCQC012W   The job card was generated with default values because the JOB keyword was missing.

Explanation: Default values were used to generate the job card because the JOB keyword was not specified in the first line of the job card.

System action: The job card was generated with default values.

User response: No action is required. To generate the job card with your own values, add the JOB keyword in the first line of the job card.

CCQC013W   The job card was generated with the default value for the programmer name because the specified programmer name exceeded 20 characters.

Explanation: Default values were used to generate the job card because the specified programmer name contained too many characters.

System action: The job card was generated with default values.

User response: No action is required. To generate the job card with your own values, add a valid programmer name in the job card. A valid programmer name is 1 - 20 characters.

CCQC014W   The job card was generated with default values because the JOB keyword was not followed by a space.

Explanation: Default values were used to generate the job card because a space did not follow the JOB keyword.

System action: The job card was generated with default values.

User response: No action is required. To generate the job card with your own values, add a space after the

JOB keyword in the job card.

CCQC015S   The *template_name* **template in the** *library_name* **metadata library contains the following file-tailoring control statement:** *statement_name***. This control statement is not valid in a** *template_type* **template.**

Explanation: The *template_type* template cannot contain the specified type of file-tailoring control statement.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQC016S   The)DOT file-tailoring control statement exceeded the number of allowed occurrences for the *template_name* **template in the** *library_name* **metadata library.**

Explanation: The)DOT file-tailoring control statement can occur only a limited number of times in the specified template.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQC017S   The nested)DOT file-tailoring control statements exceeded the number of allowed occurrences in the *template_name* **template in the** *library_name* **metadata library.**

Explanation: Nested)DOT file-tailoring control statements can occur only *number* times.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQC018S   The *template_name* **template in the** *library_name* **metadata library is not valid because it does not contain any data.**

Explanation: The specified template is missing required data.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQC019S **The** *template_name* **template in the** *library_name* **metadata library is not valid because an)ENDDOT file-tailoring control statement is missing.**

Explanation: A)ENDDOT file-tailoring control statement is required in the specified template.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQC021S **The** *template_name* **template in the** *library_name* **metadata library is not valid because the template must start with the** *parameter_name* **job card parameter.**

Explanation: The specified template must start with the specified job card parameter.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQC022S **The parameters used in a)DOT file-tailoring control statement exceeded the number of allowed parameters in the** *template_name* **template. The template is in the** *library_name* **metadata library. The error occurs in)DOT section** *section_number*.

Explanation: A)DOT file-tailoring control statement can contain only a limited number of parameters.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQC023S **The)DOT file-tailoring control statement must include the** *table-name* **table name in the** *template_name* **template. The template is in the** *library_name* **metadata library. The error occurs in)DOT section** *section_number*.

Explanation: The)DOT file-tailoring control statement is missing a required table name.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQC024S **ISPF file tailoring failed for the** *template_name* **template in the** *library_name* **metadata library.**

Explanation: An error occurred during ISPF file tailoring for the specified template.

System action: Processing stops.

User response: Review the Tools Customizer-generated trace data set and the ISPF file tailoring trace data set. To create an ISPF file tailoring trace data set, complete the following steps:

1. Run Tools Customizer until the error is about to occur.

2. Specify the ISPFTTRC command, and press Enter.

3. Issue the Tools Customizer command that issues the error.

4. Specify the ISPFTTRC command, and press Enter. The ISPF file tailoring trace data set is created. It adheres the following naming convention: *TSO_ID*.ISPFT.TRACE, where *TSO_ID* is the TSO user ID that is being used.

If the problem persists, gather the following information and contact IBM Software Support.

• A screen capture of the Tools Customizer error. Ensure that the complete error message is displayed by pressing PF1.

• The Tools Customizer trace data set. It adheres to the following naming convention: *TSO_ID*.CCQ.TRACE, where *TSO_ID* is the TSO user ID that is running Tools Customizer.

• The ISPF file tailoring trace data set.

CCQC025I **Customized jobs do not exist because they have not been generated.**

Explanation: The list of customized jobs cannot be displayed because the product has not been customized for any DB2 entries.

System action: None.

User response: Complete the steps to customize a product. Customized jobs are generated when all required product, LPAR parameters, and DB2 parameters are defined and at least one DB2 entry on which to customize the product has been selected.

CCQC026S **The value of the "customized" attribute for the** *parameter_name* **parameter in the** *library_name* **metadata library template does not match the value that was previously specified. The value is** *value_name*, **and the previously specified value is** *value_name*.

Explanation: The value for the "customized" attribute for a parameter must match the value that was previously specified.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQC027S **The** *job_name* **customization job was not found in the** *library_name* **customization library.**

Explanation: The selected customization job does not exist in the customization library.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQC028S **The** *library_name* **customization library was not found.**

Explanation: The customization library does not exist.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQC029I **The customization jobs were generated for** *Product_name***.**

Explanation: The customization jobs were generated for the specific product.

System action: None.

User response: No action is required.

CCQC030S **The customization jobs cannot be generated because at least one DB2 entry must be associated with this product.**

Explanation: The product that you are customizing requires at least one DB2 entry to be associated with it before customization jobs can be generated.

System action: None.

User response: Associate a DB2 entry with the product that you are customizing, and regenerate the jobs.

CCQC031I **The jobs were generated for the associated DB2 entries.**

Explanation: The customization jobs were generated for the DB2 entries that are associated with the product.

System action: None.

User response: No action is required.

CCQC032S **The customization jobs were not generated for** *Product_name***.**

Explanation: A severe error occurred while the jobs were being generated for the specified product.

System action: None.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQC033S **The** *customization_library_name* **has no customized jobs.**

Explanation: The specified customization library cannot be browsed or edited because it is empty.

System action: None.

User response: Generate customization jobs for the specified library, and browse or edit the library again.

CCQC034S **The specified operation is not allowed.**

Explanation: Issuing commands against customization jobs from the customization library from an ISPF browse or edit session that was started on the Finish Product Customization panel is restricted.

System action: None.

User response: To make changes to customization jobs, follow the steps for recustomization.

CCQC035E **Before you generate customization jobs, edit the product parameters to select one or more tasks or steps, and then issue the G line command or the GENERATEALL command again.**

Explanation: One or more tasks or steps must be selected before customization jobs can be generated.

System action: None.

User response: Edit the product parameters to select one or more tasks or steps. Then, issue the G line command or the GENERATEALL command again.

CCQC036E **Before you exit the Product Parameters panel, you must select one or more tasks or steps to generate customization jobs or issue the CANCEL command.**

Explanation: One or more tasks or steps must be selected to generate customization jobs or the CANCEL command must be issued before you can exit the Product Parameters panel.

System action: None.

User response: Select one or more tasks or steps, or issue the CANCEL command.

CCQD000W **The** *member_name* **environment index member is not valid. The PL/I XML parser issued the following exception warning code:** *code_number***.**

Explanation: While determining if the specified environment index member is valid, the PL/I XML parser issued an exception warning code.

System action: Processing continues.

User response: See the *Enterprise PL/I for z/OS Programming Guide* for more information about the warning.

CCQD001S **The** *member_name* **environment index member is not valid. The PL/I XML parser issued the following exception error code:** *code_number***.**

Explanation: While determining if the specified environment index member is valid, the PL/I XML parser issued an exception error code.

System action: Processing continues.

User response: See the *Enterprise PL/I for z/OS Programming Guide* for more information about the error.

CCQD002S **The XML structure of the** *member_name* **environment index member is not valid. The** *element_name* **element is unknown.**

Explanation: The specified environment index member contains an unknown element.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQD003S **The XML structure of the** *member_name* **environment index member is not valid. Content is not allowed for the** *element_name* **element, but content was found.**

Explanation: Content was found in an element that cannot contain content.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQD004S **The XML structure of the** *member_name* **environment index member is not valid. Content is required for the** *element_name* **element, but content was not found.**

Explanation: The specified element does not contain required content.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQD005S **The XML structure of the** *member_name* **environment index member is not valid. The content length for the** *element_name* **element exceeds** *maximum_number* **characters.**

Explanation: The specified element contains too many characters.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQD006S **The XML structure of the** *member_name* **environment index member is not valid. The** *element_name* **element cannot occur more than** *maximum_number* **times.**

Explanation: The specified element occurs too many times in the environment index member.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQD007S **The XML structure of the** *member_name* **environment index member is not valid. The** *element_name* **element must occur at least** *minimum_number* **times.**

Explanation: The specified element does not occur enough times in the environment index member.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQD008S **The XML structure of the** *member_name* **environment index member is not valid. The** *attribute_name* **attribute in the** *element_name* **element cannot occur more than** *maximum_number* **times.**

Explanation: The specified attribute occurs too many times in the environment index member.

System action: Processing stops.

User response: See "Gathering diagnostic

information" on page 902. Contact IBM Software Support.

CCQD009S **The XML structure of the** *member_name* **environment index member is not valid. The** *attribute_name* **attribute in the** *element_name* **element must occur at least** *minimum_number* **times.**

Explanation: The specified attribute does not occur enough times in the environment index member.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQD010S **The XML structure of the** *member_name* **environment index member is not valid. Content is not allowed for the** *attribute_name* **attribute in the** *element_name* **element, but content was found.**

Explanation: Content was found in an attribute that cannot contain content. The name of the attribute and the name of the element that contains it are indicated in the message text.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQD011S **The XML structure of the** *member_name* **environment index member is not valid. Content is required for the** *attribute_name* **attribute in the** *element_name* **element, but content was not found.**

Explanation: An attribute does not contain required content. The name of the attribute and the name of the element that contains it are indicated in the message text.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQD012S **The XML structure of the** *member_name* **environment index member is not valid. The content length for the** *element_name* **element exceeds** *maximum_number* **characters.**

Explanation: An element contains too many characters. The name of the element and the maximum

number of allowed characters are indicated in the message text.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQD013S **The XML structure of the** *member_name* **environment index member is not valid. The** *attribute_name* **attribute in the** *element_name* **element is unknown.**

Explanation: The environment index member contains an unknown attribute. The name of the unknown attribute and the name of the element that contains it are indicated in the message text.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQD050S **The following LPAR serial number is duplicated in the environment index member:** *serial_number*.

Explanation: The environment index member contains duplicate LPAR serial numbers. The duplicate serial number is indicated in the message text.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQD051S **The following DB2 serial number is duplicated in the environment index member:** *serial_number*.

Explanation: The environment index member contains duplicate DB2 serial numbers. The duplicate serial number is indicated in the message text.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQD052S **The following DB2 group attach name is duplicated in the environment index member:** *group_attach_name*.

Explanation: The environment index member contains duplicate group attach names.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQD053S	**The reference to the following DB2 subsystem for a DB2 group attach name is duplicated in the environment index member:** *subsystem_ID***.**

Explanation: The environment index member contains duplicate references to a DB2 subsystem for a DB2 group attach name.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQD054S	**The reference to the following DB2 subsystem for the** *LPAR_name* **LPAR is duplicated in the environment index member:** *subsystem_ID***.**

Explanation: The environment index member contains duplicate references to a DB2 subsystem for an LPAR. The duplicate subsystem ID is indicated in the message text.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQD055S	**The following DB2 group attach name was not found in the environment index member:** *group_attach_name***.**

Explanation: A group attach name that is referenced by a DB2 member does not exist in the environment index member.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQD056S	**The following LPAR was not found in the environment index member:** *LPAR_name***.**

Explanation: The LPAR does not exist in the environment index member.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQD057S	**The following LPAR is duplicated in the environment index member:** *LPAR_name***.**

Explanation: The environment index member contains duplicate LPARs. The name of the duplicate LPAR name is indicated in the message text.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQD100W	**The** *member_name* **product index member is not valid. The PL/I XML parser issued the following exception warning code:** *code_number***.**

Explanation: While determining if the product index member is valid, the PL/I XML parser issued the specified exception warning code.

System action: Processing continues.

User response: See the *Enterprise PL/I for z/OS Programming Guide* for more information about the specified exception warning code.

CCQD101S	**The** *member_name* **product index member is not valid. The PL/I XML parser issued the following exception error code:** *code_number***.**

Explanation: While determining if the product index member is valid, the PL/I XML parser issued the specified exception error code.

System action: Processing stops.

User response: See the *Enterprise PL/I for z/OS Programming Guide* for more information about the specified exception error code. Ensure that the Tools Customizer data store data set DCB is the same as the sample SCCQSAMP(CCQCDATS) data set DCB.

CCQD102S	**The XML structure of the** *member_name* **product index member is not valid. The** *element_name* **element is unknown.**

Explanation: The specified product index member contains an unknown element.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQD103S	**The XML structure of the** *member_name* **product index member is not valid. Content is not allowed for the** *element_name* **element, but content was found.**

Explanation: Content was found for an element that cannot contain content.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQD104S   **The XML structure of the** *member_name* **product index member is not valid. Content is required for the** *element_name* **element, but content was not found.**

Explanation: The specified element does not contain required content.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQD105S   **The XML structure of the** *member_name* **product index member is not valid. The content length for the** *element_name* **element exceeds** *maximum_number* **characters.**

Explanation: The specified element contains too many characters.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQD106S   **The XML structure of the** *member_name* **product index member is not valid. The** *element_name* **element cannot occur more than** *maximum_number* **times.**

Explanation: The specified element occurs too many times in the product index member.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQD107S   **The XML structure of the** *member_name* **product index member is not valid. The** *element_name* **element must occur at least** *minimum_number* **times.**

Explanation: The specified element does not occur enough times in the product index member.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQD108S   **The XML structure of the** *member_name* **product index member is not valid. The** *attribute_name* **attribute in the** *element name* **element cannot occur more than** *maximum_number* **times.**

Explanation: An attribute occurs too many times. The

name of the attribute and the element that contains it are indicated in the message text.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQD109S   **The XML structure of the** *member_name* **product index member is not valid. The** *attribute_name* **attribute in the** *element_name* **element must occur at least** *minimum_number* **times.**

Explanation: The specified attribute does not occur enough times in the product index member.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQD110S   **The XML structure of the** *member_name* **product index member is not valid. Content is not allowed for the** *attribute_name* **attribute in the** *element_name* **element, but content was found.**

Explanation: An attribute cannot contain content. The name of the attribute and the element that contains it are indicated in the message text.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQD111S   **The XML structure of the** *member_name* **product index member is not valid. Content is required for the** *attribute_name* **attribute in the** *element_name* **element, but content was not found.**

Explanation: An attribute requires content. The name of the attribute and the name of the element that contains it are indicated in the message text.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQD112S   **The XML structure of the** *member_name* **product index member is not valid. The content length for the** *element_name* **element exceeds** *maximum_number* **characters.**

Explanation: The specified element contains too many characters.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQD113S **The XML structure of the** *member_name* **product index member is not valid. The** *attribute_name* **attribute in the** *element_name* **element is unknown.**

Explanation: The specified attribute in the product index member is unknown.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQD118S **The content of the** *member_name* **product index member is not valid. The** *configuration_ID* **configuration ID for the** *configuration-name* **configuration name is not unique.**

Explanation:

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQD120S **The content of the** *member_name* **product index member is not valid. The pack ID** *pack_ID* **that is referenced by product prefix** *product_prefix* **in the metadata library** *library_name* **could not be found.**

Explanation: The specified pack ID could not be found in the metadata library.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQD121I **The specified pack contains the** *component_name***, which was previously specified as a stand-alone product.**

Explanation: The specified component of the pack was previously specified as a stand-alone product.

System action: None.

User response: No action is required.

CCQD122I **The specified component metadata library was previously specified as part of the** *pack_name***.**

Explanation: The specified metadata library for the component was previously specified as part of a pack.

System action: None.

User response: No action is required.

CCQD123E **The customization library name** *library_name* **is being used by another product or component. Specify another customization library qualifier on the Tools Customizer Settings panel.**

Explanation: A different product or component is using the specified customization library.

System action: None.

User response: Specify another customization library qualifier on the Tools Customizer Settings panel.

CCQD300W **The** *member_name* **product environment member is not valid. The PL/I XML parser issued the following exception warning code:** *code_number***.**

Explanation: While determining if the product environment member is valid, the PL/I XML parser issued the specified exception warning code.

System action: Processing continues.

User response: See the *Enterprise PL/I for z/OS Programming Guide* for more information about the specified exception warning code.

CCQD301S **The** *member_name* **product environment member is not valid. The PL/I XML parser issued the following exception error code:** *code_number***.**

Explanation: While determining if the product environment member is valid, the PL/I XML parser issued the specified exception error code.

System action: Processing stops.

User response: See the *Enterprise PL/I for z/OS Programming Guide* for more information about the specified exception error code.

CCQD302S **The XML structure of the** *member_name* **product environment member is not valid. The** *element_name* **element is unknown.**

Explanation: The specified product environment member contains an unknown element.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQD303S **The XML structure of the** *member_name* **product environment member is not valid. Content is not allowed for the** *element_name* **element, but content was found.**

Explanation: Content was found for an element that cannot contain content.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQD304S **The XML structure of the** *member_name* **product environment member is not valid. Content is required for the** *element_name* **element, but content was not found.**

Explanation: The specified element does not contain required content.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQD305S **The XML structure of the** *member_name* **product environment member is not valid. The content length for the** *element_name* **element exceeds** *maximum_number* **characters.**

Explanation: The specified element contains too many characters.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQD306S **The XML structure of the** *member_name* **product environment member is not valid. The** *element_name* **element cannot occur more than** *maximum_number* **times.**

Explanation: The specified element occurs too many times in the product environment member.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQD307S **The XML structure of the** *member_name* **product environment member is not valid. The** *element_name* **element must occur at least** *minimum_number* **times.**

Explanation: The specified element does not occur enough times in the product environment member.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQD308S **The XML structure of the** *member_name* **product environment member is not valid. The** *attribute_name* **attribute in the** *element name* **element cannot occur more than** *maximum_number* **times.**

Explanation: The specified attribute occurs too many times. The name of the attribute and the element that contains it are indicated in the message text.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQD309S **The XML structure of the** *member_name* **product environment member is not valid. The** *attribute_name* **attribute in the** *element_name* **element must occur at least** *minimum_number* **times.**

Explanation: The specified attribute does not occur enough times in the product environment member.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQD310S **The XML structure of the** *member_name* **product environment member is not valid. Content is not allowed for the** *attribute_name* **attribute in the** *element_name* **element, but content was found.**

Explanation: The specified attribute cannot contain content. The name of the attribute and the element that contains it are indicated in the message text.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQD311S   **The XML structure of the** *member_name* **product environment member is not valid. Content is required for the** *attribute_name* **attribute in the** *element_name* **element, but content was not found.**

Explanation: The specified attribute requires content. The name of the attribute and the name of the element that contains it are indicated in the message text.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQD312S   **The XML structure of the** *member_name* **product environment member is not valid. The content length for the** *element_name* **element exceeds** *maximum_number* **characters.**

Explanation: The specified element contains too many characters.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQD313S   **The XML structure of the** *member_name* **product environment member is not valid. The** *attribute_name* **attribute in the** *element_name* **element is unknown.**

Explanation: The specified attribute in the product environment member is unknown.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQD350I   **The** *subsystem_ID* **DB2 subsystem is associated with this product.**

Explanation: The specified DB2 subsystem was added and saved in the Tools Customizer data store for the product to be customized.

System action: Processing continues.

User response: No action is required.

CCQD351I   **The** *member_name* **DB2 member for the** *group_attach_name* **DB2 group attach name is associated with this product.**

Explanation: The specified DB2 member for the group attach name was added and saved in the Tools Customizer data store for the product to be customized.

System action: Processing continues.

User response: No action is required.

CCQD352I   **The** *group_attach_name* **DB2 group attach name is associated with this product.**

Explanation: The specified DB2 group attach name was added and saved in the Tools Customizer data store for the product to be customized.

System action: Processing continues.

User response: No action is required.

CCQD353E   **The** *subsystem_ID* **DB2 subsystem is already associated with this product.**

Explanation: The specified DB2 subsystem cannot be added for the product to be customized because it already exists in the product environment in the data store.

System action: None.

User response: Ensure that the DB2 subsystem is specified correctly. If the problem persists, contact IBM Software Support.

CCQD354E   **The** *member_name* **DB2 member for the** *group_attach_name* **DB2 group attach name is already associated with this product.**

Explanation: The specified DB2 member for the group attach name cannot be added for the product to be customized because it already exists in the product environment in the data store.

System action: None.

User response: Ensure that the DB2 group attach name is specified correctly. If the problem persists, contact IBM Software Support.

CCQD355E   **The** *group_attach_name* **DB2 group attach name is already associated with this product.**

Explanation: The specified DB2 group attach name cannot be added for the product to be customized because it already exists in the product environment in the data store.

System action: Processing stops.

User response: Ensure that the DB2 group attach name is specified correctly. If the problem persists, contact IBM Software Support.

CCQD356S **The** *library_name* **metadata library is already associated with the maximum number of allowed DB2 entries for this product.**

Explanation: The specified metadata library cannot be associated with more DB2 entries because it is already associated with the number of DB2 entries that are allowed.

System action: Processing stops.

User response: Delete an associated DB2 entry, and associate the specified library with another DB2 entry again.

CCQD357I **The** *subsystem_ID* **DB2 subsystem is unassociated with this product.**

Explanation: The specified DB2 SSID was unassociated with the product that you are customizing.

System action: Processing continues.

User response: No action is required.

CCQD358I **The** *member_name* **DB2 member for the** *group_attach_name* **DB2 group attach name is unassociated with this product.**

Explanation: The specified DB2 member for the DB2 group attach name was unassociated with the product that you are customizing.

System action: Processing continues.

User response: No action is required.

CCQD359I **The** *group_attach_name* **DB2 group attach name is unassociated with this product.**

Explanation: The specified DB2 group attach name was unassociated with the product that you are customizing.

System action: Processing continues.

User response: No action is required.

CCQD360S **The** *library_name* **metadata library is not associated with the specified DB2 subsystem** *subsystem_ID*.

Explanation: The specified DB2 subsystem and metadata library are not associated with each other.

System action: None.

User response: Ensure that the DB2 subsystem and the metadata library are associated. If the problem persists, contact IBM Software Support.

CCQD361S **The** *library_name* **metadata library is not associated with the specified DB2 data sharing group member** *member_name* **for the** *group_attach_name* **DB2 group attach name.**

Explanation: The specified DB2 data sharing group member for the group attach name and metadata library are not associated with each other.

System action: None.

User response: Ensure that the DB2 data sharing group member for the group attach name and the metadata library are associated. If the problem persists, contact IBM Software Support.

CCQD362S **The** *library_name* **metadata library is not associated with the specified** *group_attach_name* **DB2 group attach name.**

Explanation: The specified DB2 group attach name and metadata library are not associated with each other.

System action: None.

User response: Ensure that the DB2 group attach name and the metadata library are associated. If the problem persists, contact IBM Software Support.

CCQD400W **The customization parser issued the** *code_number* **warning code while it parsed the product customization member** *member_name*. **See the PL/I programming guide for more information about this XML parser continuable exception code.**

Explanation: While determining if the specified member is valid, the PL/I XML parser issued an exception warning code.

System action: Processing stops.

User response: See the *Enterprise PL/I for z/OS Programming Guide* for more information about the warning.

CCQD401S **The customization parser issued the** *code_number* **error code while it parsed the product customization member** *member_name*. **See the PL/I programming guide for more information about this XML parser terminating exception code.**

Explanation: While determining if the specified member is valid, the PL/I XML parser issued an exception error code.

System action: Processing stops.

User response: See the *Enterprise PL/I for z/OS*

Programming Guide for more information about the error.

CCQD500W The *data_set_name* **data store data set was not found.**

Explanation: Tools Customizer could not find the specified data store data set.

System action: None.

User response: No action is required.

CCQD501W The *data_set_name* **data store data set was not found, so it was created.**

Explanation: Tools Customizer created the specified data set because it could not be found.

System action: None.

User response: No action is required.

CCQD502E The *data_set_name* **data store data set is not writable.**

Explanation: Tools Customizer cannot write to the specified data set.

System action: None.

User response: Ensure that the data set is writable.

CCQD503E The *data_set_name* **data store data set could not be opened with the** *disposition_type* **disposition.**

Explanation: Tools Customizer could not open the data set with the specified disposition.

System action: Processing stops.

User response: Ensure that you have WRITE authority access to this data set.

CCQD504E The *data_set_name* **data store data set could not be opened with the** *option_name* **option.**

Explanation: Tools Customizer could not open the data set with the specified option.

System action: Processing stops.

User response: Ensure that you have WRITE authority access to this data set.

CCQD505E The *data_set_name* **data store data set could not be created.**

Explanation: Tools Customizer could not create the specified data set.

System action: Processing stops.

User response: Ensure that you have the authority to create data sets and that the DASD is not full.

CCQD510I The DB2 SSID and DB2 group attach name were created.

Explanation: The DB2 SSID and DB2 group attach name were created and saved in the data store.

System action: None.

User response: No action is required.

CCQD511E The DB2 entry already exists in the list of DB2 entries to be associated.

Explanation: The DB2 entry cannot be added because it already exists in the list of DB2 entries to be associated.

System action: None.

User response: Specify a different DB2 entry.

CCQD512S An error occurred while a DB2 entry was being created.

Explanation: A severe error occurred while a DB2 entry was being created.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQD513E The specified DB2 entry already exists and is associated with the current product on the Customizer Workplace panel.

Explanation: The DB2 entry cannot be added because it already exists, and it is already associated with the product to be customized.

System action: None.

User response: Press F3 to go to the Customizer Workplace panel to see the DB2 entry, or specify a different DB2 entry.

CCQD514E A value is required for a DB2 subsystem, a DB2 group attach name, or both before they can be created.

Explanation: Required information is missing. A DB2 subsystem, a DB2 group attach name, or both must be specified.

System action: None.

User response: Specify a DB2 subsystem, a DB2 group attach name, or both.

CCQD515E   The specified DB2 entry already exists in the list of DB2 entries and is already associated with the current product.

Explanation: The DB2 entry has already been created and associated with the product that you want to customize.

System action: None.

User response: Specify a different DB2 entry.

CCQD516E   The specified DB2 entry already exists in the list of DB2 entries on the Associate DB2 Entry with Product panel but is not associated with the current product.

Explanation: The DB2 entry exists, but it must be associated with the product to be customized.

System action: None.

User response: On the Customizer Workplace panel, issue the ASSOCIATE command to associate the DB2 entry with the product.

CCQD517S   An error occurred while a DB2 entry was being copied.

Explanation: A severe error occurred while a DB2 entry was being copied

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQD518E   A value is required for a DB2 subsystem, a DB2 group attach name, or both before they can be copied.

Explanation: Required information is missing. A DB2 subsystem, a DB2 group attach name, or both must be specified.

System action: None.

User response: Specify a DB2 subsystem, a DB2 group attach name, or both.

CCQD519I   The DB2 entry was copied.

Explanation: The DB2 entry was copied and saved in the Tools Customizer data store.

System action: None.

User response: No action is required.

CCQD520S   The DB2 entry was copied to the list of DB2 entries but was not associated because the product is already associated with the allowed number of DB2 entries.

Explanation: The DB2 entry was not completely copied because a product can be associated with only 1200 DB2 entries.

System action: Processing stops.

User response: Remove a DB2 entry from the list, and copy the specified DB2 entry again.

CCQD521E   *Line_command* is not a valid line command.

Explanation: The specified line command is not valid. Valid line commands are on the panel.

System action: Processing stops.

User response: Specify a valid line command.

CCQD522E   The *subsystem_ID* DB2 subsystem ID occurs more than once in the list. Each row must be unique.

Explanation: The specified DB2 subsystem ID can be used only once.

System action: Processing stops.

User response: Specify a different DB2 subsystem ID.

CCQD523E   The *group_attach_name* DB2 group attach name occurs more than once in the list. Each row must be unique.

Explanation: The specified DB2 group attach name can be used only once.

System action: Processing stops.

User response: Specify a different DB2 group attach name.

CCQD524E   The *member_name* DB2 member for the DB2 group attach name occurs more than once in the list. Each row must be unique.

Explanation: The specified DB2 member for the DB2 group attach name can be used only once.

System action: Processing stops.

User response: Specify a different DB2 member for the DB2 group attach name.

| **CCQD525I   The DB2 entries were created.**

| **User response:** No action is required.

| **CCQD526E   The** *subsystem_ID* **DB2 subsystem ID**
| **occurs more than once in the list. Each**
| **DB2 subsystem ID must be unique.**

| **Explanation:** The specified DB2 subsystem ID can be
| used only once.

| **System action:** Processing stops.

| **User response:** Specify a different DB2 subsystem ID.

| **CCQD527I   DB2 group attach names cannot be**
| **created during the copy process.**

| **Explanation:** The ability to create DB2 group attach
| names is not available during the copy process.

| **System action:** None.

| **User response:** Create DB2 group attach names by
| issuing the CREATE command on the Customizer
| Workplace panel.

| **CCQD528E   The** *metadata_library* **metadata library is**
| **already associated with** *number* **DB2**
| **entries. The maximum number of**
| **associated DB2 entries for this metadata**
| **library is 256.**

| **Explanation:** A metadata library can be associated
| with a maximum of 256 DB2 entries. The specified
| metadata library is already associated with 256.

| **System action:** Processing stops.

| **User response:** Remove an existing association
| between the specified metadata library and a DB2
| entry, and associate the specified the metadata library
| with another entry.

| **CCQD529I   At least one row is required.**

CCQD560E   The *subsystem_ID* **DB2 subsystem**
                already exists and is associated with the
                current product on the Customizer
                Workplace panel.

Explanation: The specified DB2 subsystem exists and
is associated with the product that you are customizing.

System action: None.

User response: Specify another DB2 subsystem.

CCQD561E   The *member_name* **DB2 member for the**
                group_attach_name **DB2 group attach**
                name already exists and is associated
                with the current product on the
                Customizer Workplace panel.

Explanation: The specified DB2 data sharing group for
the DB2 group attach namer exists and is associated
with the product that you are customizing.

System action: None.

User response: Specify another DB2 subsystem.

CCQD562E   The *group_attach_name* **DB2 group attach**
                name already exists and is associated
                with the current product on the
                Customizer Workplace panel.

Explanation: The specified DB2 group attach name
exists and is associated with the product that you are
customizing. The subsystem is in the table on the
Customizer Workplace panel.

System action: None.

User response: Specify another DB2 group attach
name.

CCQD563E   A value is required for a DB2
                subsystem, a DB2 group attach name, or
                both before they can be created.

Explanation: A DB2 subsystem, a DB2 group attach
name, or both are not specified so one or both of them
cannot be created.

System action: None.

User response: Specify a value for the DB2 subsystem,
the DB2 group attach name, or both.

CCQD565E   The *subsystem_ID* **DB2 subsystem**
                already exists in the list of DB2 entries
                and is already associated with the
                current product.

Explanation: The specified subsystem is already
associated.

System action: None.

User response: Specify a different DB2 subsystem.

CCQD566E   The *member_name* **DB2 member for the**
                group_attach_name **DB2 group attach**
                name already exists in the list of DB2
                entries and is already associated with
                the current product.

Explanation: The specified DB2member is already
associated.

System action: None.

User response: Specify a different DB2 member.

CCQD567E The *group_attach_name* **DB2 group attach name already exists in the list of DB2 entries and is already associated with the current product.**

Explanation: The specified DB2 group attach name is already associated.

System action: None.

User response: Specify another DB2 group attach name.

CCQD568I To customize *product_name*, at least one **DB2 entry must be associated with this product.**

Explanation: The specified product requires at least one associated DB2 entry.

System action: None.

User response: To continue the customization process for the specified product, associate one or more DB2 entries with it.

CCQD569I To customize the *product_name* **product configuration, at least one DB2 entry must be associated with this configuration.**

Explanation: The configuration for the specified product requires at least one associated DB2 entry.

System action: None.

User response: To continue the customization process for the configuration of the specified product, associate one or more DB2 entries with the configuration.

CCQD577W The *mode_name* **DB2 mode of the** *subsystem_ID* **DB2 subsystem is not supported by the product.**

Explanation: The product does not support the specified DB2 mode.

System action: None.

User response: Specify a supported DB2 mode.

CCQD578W The *mode_name* **DB2 mode of the** *member_name* **DB2 member for the DB2 group is not supported by the product.**

Explanation: The product does not support the specified DB2 mode.

System action: None.

User response: Specify a supported DB2 mode.

CCQD579W The *mode_name* **DB2 mode of the** *group_name* **DB2 group attach name is not supported by the product.**

Explanation: The product does not support the specified DB2 mode.

System action: None.

User response: Specify a supported DB2 mode.

CCQD580S The *subsystem_ID* **DB2 subsystem was copied to the list of DB2 entries but was not associated because the product is already associated with the allowed number of DB2 entries.**

Explanation: The copied DB2 subsystem was not associated with the product because the product is associated with the maximum number of DB2 entries.

System action: None.

User response: Remove an associated DB2 entry and associate the product with the copied DB2 subsystem.

CCQD581S The *member_name* **DB2 member for the** *group_attach_name* **DB2 group attach name was copied to the list of DB2 entries but was not associated because the product is already associated with the allowed number of DB2 entries.**

Explanation: The copied DB2 member for the DB2 group attach name was not associated with the product because the product is associated with the maximum number of DB2 entries.

System action: None.

User response: Remove an associated DB2 entry and associate the product with the copied DB2 member.

CCQD582S The *group_attach_name* **DB2 group attach name was copied to the list of DB2 entries but was not associated because the product is already associated with the allowed number of DB2 entries.**

Explanation: The copied DB2 group attach name was not associated with the product because the product is associated with the maximum number of DB2 entries.

System action: None.

User response: Remove an associated DB2 entry and associate the product with the copied DB2 group attach name.

CCQD584I The *member_name* DB2 member for the *group_attach_name* DB2 group attach name is copied to the *subsystem_ID* DB2 subsystem.

Explanation: The specified DB2 member was copied.

System action: None.

User response: No action is required.

CCQD585I The *group_attach_name* DB2 group attach name cannot be copied because a DB2 member is required.

Explanation: The specified DB2 group attach name was not copied because a DB2 member was missing.

System action: None.

User response: No action is required.

CCQD586S The current LPAR is *LPAR_name*, but the data store contains information about the *LPAR_name* LPAR. You must use the *LPAR_name* LPAR to customize the product.

Explanation: The LPAR that is stored in the data store data set must be used to customize the product.

System action: Processing stops.

User response: Use the LPAR that is stored in the data store data set.

CCQD587W The *level_number* DB2 level of the *subsystem_name* DB2 subsystem is not supported by the product.

Explanation: The product does not support the specified DB2 level.

System action: Processing continues.

User response: Specify a supported level of DB2.

CCQD588W The *level_number* DB2 level of the *member_name* DB2 member of the *group_name* DB2 group is not supported by the product.

Explanation: The product does not support the specified DB2 level.

System action: Processing continues.

User response: Specify a supported level of DB2.

CCQD589W The *level_number* DB2 level of the *group_name* DB2 group attach name is not supported by the product.

Explanation: The product does not support the specified DB2 level.

System action: Processing continues.

User response: Specify a supported level of DB2.

CCQD593I The *subsystem_ID* DB2 subsystem was deleted.

User response: No action is required.

CCQD594I The *member_name* DB2 for the *group_attach_name* DB2 group attach name was deleted.

User response: No action is required.

CCQD595I The *group_attach_name* DB2 group attach name was deleted.

User response: No action is required.

CCQD596E The *subsystem_ID* DB2 subsystem was not deleted.

Explanation: An internal error occurred while the specified DB2 subsystem was being deleted.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQD597E The *member_name* DB2 member for the *group_attach_name* DB2 group attach name was not deleted.

Explanation: An internal error occurred while the specified DB2 member was being deleted.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQD598E The *group_attach_name* DB2 group attach name was not deleted.

Explanation: An internal error occurred while the specified DB2 group attach name was being deleted.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQD600W The *member_name* product customization member is not valid. The PL/I XML parser issued the following exception warning code: *code_number*.

Explanation: While determining if the XML structure of the product customization member is valid, the PL/I

XML parser issued an exception warning code.

System action: Processing continues.

User response: See the *Enterprise PL/I for z/OS Programming Guide* for more information about the exception warning code.

CCQD601S   **The** *member_name* **product customization member is not valid. The PL/I XML parser issued the following exception error code:** *code_number*.

Explanation: While determining if the XML structure of the product customization member is valid, the PL/I XML parser issued an exception error code.

System action: Processing stops.

User response: See the *Enterprise PL/I for z/OS Programming Guide* for more information about the exception error code.

CCQD602S   **The XML structure of the** *member_name* **product customization member is not valid. The** *element_name* **element is unknown.**

Explanation: The data store member contains an unknown element.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQD603S   **The XML structure of the** *member_name* **product customization member is not valid. Content is not allowed for the** *element_name* **element, but content was found.**

Explanation: The specified element cannot contain content.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQD604S   **The XML structure of the** *member_name* **product customization member is not valid. Content is required for the** *element_name* **element, but content was not found.**

Explanation: The specified element is missing required content.

System action: Processing stops.

User response: See "Gathering diagnostic

information" on page 902. Contact IBM Software Support.

CCQD605S   **The XML structure of the** *member_name* **product customization member is not valid. The content length for the** *element_name* **element exceeds** *maximum_number* **characters.**

Explanation: The specified element contains too many characters.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQD606S   **The XML structure of the** *member_name* **product customization member is not valid. The** *element_name* **element cannot occur more than** *maximum_number* **times.**

Explanation: The specified element occurs too many times.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQD607S   **The XML structure of the** *member_name* **product customization member is not valid. The** *element_name* **element must occur at least** *minimum_number* **times.**

Explanation: The specified element does not occur enough times.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQD608S   **The XML structure of the** *member_name* **product customization member is not valid. The** *attribute_name* **attribute in the** *element_name* **element cannot occur more than** *maximum_number* **times.**

Explanation: The specified attribute occurs too many times.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQD609S **The XML structure of the** *member_name* **product customization member is not valid. The** *attribute_name* **attribute in the** *element_name* **element must occur at least** *minimum_number* **times.**

Explanation: The specified attribute does not occur enough times.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQD610S **The XML structure of the** *member_name* **product customization member is not valid. Content is not allowed for the** *attribute_name* **attribute in the** *element_name* **element, but content was found.**

Explanation: The specified attribute cannot contain content.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQD611S **The XML structure of the** *member_name* **product customization member is not valid. Content is required for the** *attribute_name* **attribute in the** *element_name* **element, but content was not found.**

Explanation: The specified attribute does not contain required content.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQD612S **The XML structure of the** *member_name* **product customization member is not valid. The content length for the** *element_name* **element exceeds** *maximum_number* **characters.**

Explanation: The specified element contains too many characters.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQD613S **The XML structure of the** *member_name* **product customization member is not valid. The** *attribute_name* **attribute in the** *element_name* **element is unknown.**

Explanation: The specified attribute in the data store member is unknown.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQD614S **The content of the** *member_name* **product customization member is not valid. The value of the** *element_name* **element is not valid. The value is** *value_name***.**

Explanation: The specified value is not valid.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQD700W **The** *member_name* **DB2 data member is not valid. The PL/I XML parser issued the following exception warning code:** *code_number***.**

Explanation: While determining if the XML structure of the DB2 data member is valid, the PL/I XML parser issued an exception warning code.

System action: Processing continues.

User response: See the *Enterprise PL/I for z/OS Programming Guide* for more information about the exception warning code.

CCQD701S **The** *member_name* **DB2 data member is not valid. The PL/I XML parser issued the following exception error code:** *code_number***.**

Explanation: While determining if the XML structure of the DB2 data member is valid, the PL/I XML parser issued an exception error code.

System action: Processing continues.

User response: See the *Enterprise PL/I for z/OS Programming Guide* for more information about the exception error code.

CCQD750W **The** *value_number* **value in the DB2 parameter** *parameter_name* **was skipped because only** *maximum_number* **values are allowed.**

Explanation: The specified value was skipped because

it exceeds the number of allowed values in the DB2 parameter.

System action: Processing continues.

User response: No action is required. To stop this message from being issued, remove the extra values from the DB2 parameter.

CCQD800W **The** *member_name* **LPAR data member is not valid. The PL/I XML parser issued the following exception warning code:** *code_number***.**

Explanation: While determining if the XML structure of the LPAR data member is valid, the PL/I XML parser issued an exception warning code.

System action: Processing continues.

User response: See the *Enterprise PL/I for z/OS Programming Guide* for more information about the exception warning code.

CCQD801S **The** *member_name* **LPAR data member is not valid. The PL/I XML parser issued the following exception error code:** *code_number***.**

Explanation: While determining if the XML structure of the LPAR data member is valid, the PL/I XML parser issued an exception error code.

System action: Processing continues.

User response: See the *Enterprise PL/I for z/OS Programming Guide* for more information about the exception error code.

CCQD850W **The** *value_number* **value in the LPAR parameter** *parameter_name* **was skipped because only** *maximum_number* **values are allowed.**

Explanation: The specified value was skipped because it exceeds the number of allowed values in the LPAR parameter.

System action: Processing continues.

User response: No action is required. To stop this message from being issued, remove the extra values from the LPAR parameter.

CCQD851I **The** *subsystem_ID* **DB2 subsystem is copied to the** *member_name* **DB2 member for the** *group_attach_name* **DB2 group attach name.**

User response: No action is required.

CCQD852I **The** *member_name* **DB2 member for the** *group_attach_name* **DB2 group attach name is copied to the** *member_name* **DB2 member for the** *group_attach_name* **DB2 group attach name.**

User response: No action is required.

CCQD854I **The** *member_name* **DB2 member for the** *group_attach_name* **DB2 group 'attach name is copied to multiple DB2 entries.**

User response: No action is required.

CCQD900W **The** *member_name* **product data member is not valid. The PL/I XML parser issued the following exception warning code:** *code_number***.**

Explanation: While determining if the XML structure of the product data member is valid, the PL/I XML parser issued an exception warning code.

System action: Processing continues.

User response: See the *Enterprise PL/I for z/OS Programming Guide* for more information about the exception warning code.

CCQD901S **The** *member_name* **product data member is not valid. The PL/I XML parser issued the following exception error code:** *code_number***.**

Explanation: While determining if the XML structure of the product data member is valid, the PL/I XML parser issued an exception error code.

System action: Processing continues.

User response: See the *Enterprise PL/I for z/OS Programming Guide* for more information about the exception warning code.

CCQD950W **The** *value_number* **value in the product parameter** *parameter_name* **was skipped because only** *maximum_number* **values are allowed.**

Explanation: The specified value was skipped because it exceeds the number of allowed values in the product parameter.

System action: Processing continues.

User response: No action is required. To stop this message from being issued, remove the extra values from the product parameter.

| **CCQD960I** | **The** *subsystem_ID* **DB2 subsystem was changed to the** *member_name* **DB2 member for the** *group_attach_name* **DB2 group attach name.**

| **User response:** No action is required.

| **CCQD961I** | **The** *member_name* **DB2 member for the** *group_attach_name* **DB2 group attach name was changed to the** *subsystem_ID* **DB2 subsystem.**

| **User response:** No action is required.

| **CCQD962I** | **The** *member_name* **DB2 member for the** *group_attach_name* **DB2 group attach name was changed to the** *member_name* **DB2 member for the** *group_attach_name* **DB2 group attach name.**

| **User response:** No action is required.

| **CCQD963E** | **The DB2 group attach name cannot be blank when the DB2 subsystem ID is blank.**

| **Explanation:** A DB2 group attach name, DB2 subsystem ID, or both must be specified.

| **System action:** Processing stops.

| **User response:** Specify a DB2 group attach name, DB2 subsystem ID, or both.

CCQE000S | **The specified message field name or message** *message_ID* **was not found.**

Explanation: An error occurred while displaying a message field name or the specified message.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQE001E | **An incorrect trace level was specified. Valid trace levels are 0 - 4.**

Explanation: A wrong trace level was specified. Valid trace levels are 0 - 4.

System action: Processing stops.

User response: Specify a valid trace level 0 - 4.

CCQH001W | **The specified option** *option_name* **is not valid.**

Explanation: The option that was specified is not a valid option on the panel.

System action: Tools Customizer stops.

User response: Specify a valid option on the panel.

CCQH006W | **Before you customize a product, verify your user settings.**

Explanation: The user settings must be verified before a product can be customized.

System action: Tools Customizer stops.

User response: Verify the user settings.

CCQH007E | **Check the user settings. One or more current values are not valid.**

Explanation: One or more of the values in the user settings is not valid.

System action: Tools Customizer stops.

User response: Ensure that the specified values for the user settings are valid.

CCQH008W | **Before you use Tools Customizer, you must select option 0 to verify your user settings.**

Explanation: The user settings must be changed before a product can be customized.

System action: Tools Customizer stops.

User response: Change the user settings.

CCQH009E | **You must select option 0 to change your user settings.**

Explanation: User settings must be changed before a product can be customized.

System action: Tools Customizer stops.

User response: Change the user settings.

CCQI000W | **The XML structure of the** *member_name* **DB2 parameter metadata member is not valid. The PL/I XML parser issued the following exception warning code:** *code_number***.**

Explanation: While determining if the DB2 parameter metadata member is valid, the PL/I XML parser issued an exception warning code.

System action: Processing continues.

User response: See the *Enterprise PL/I for z/OS Programming Guide* for more information about the exception warning code.

CCQI001S **The XML structure of the** *member_name* **DB2 parameter metadata member is not valid. The PL/I XML parser issued the following exception error code:** *code_number***.**

Explanation: While determining if the DB2 parameter metadata member is valid, the PL/I XML parser issued an exception error code.

System action: Processing stops.

User response: See the *Enterprise PL/I for z/OS Programming Guide* for more information about the exception warning code.

CCQI002S **The XML structure of the** *member_name* **DB2 parameter metadata member is not valid. The** *element name* **element is unknown.**

Explanation: The specified element in the DB2 parameter metadata member is unknown.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI003S **The XML structure of the** *member_name* **DB2 parameter metadata member is not valid. Content is not allowed for the** *element_name* **element, but content was found.**

Explanation: The specified element cannot contain content.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI004S **The XML structure of the** *member_name* **DB2 parameter metadata member is not valid. Content is required for the** *element_name* **element, but content was not found.**

Explanation: The specified element requires content.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI005S **The XML structure of the** *member_name* **DB2 parameter metadata member is not valid. The content length for the** *element_name* **element cannot exceed** *maximum_number* **characters.**

Explanation: The specified element contains too many characters.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI006S **The XML structure of the** *member_name* **DB2 parameter metadata member is not valid. The content length for the** *element_name* **element must be at least** *minimum_number* **characters.**

Explanation: The specified element does not contain enough characters.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI007S **The XML structure of the** *member_name* **DB2 parameter metadata member is not valid. The** *element_name* **element must occur at least** *minimum_number* **times.**

Explanation: The specified element does not occur enough times.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI008S **The XML structure of the** *member_name* **DB2 parameter metadata member is not valid. The** *attribute_name* **attribute in the** *element_name* **element cannot occur more than** *maximum_number* **times.**

Explanation: The specified attribute occurs too many times.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI009S   The XML structure of the *member_name* DB2 parameter metadata member is not valid. The *attribute_name* attribute in the *element_name* element must occur at least *minimum_number* times.

Explanation: The specified attribute did not occur enough times.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI010S   The XML structure of the *member_name* DB2 parameter metadata member is not valid. Content is not allowed for the *attribute_name* attribute in the *element_name* element, but content was found.

Explanation: The specified attribute cannot have content.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI011S   The XML structure of the *member_name* DB2 parameter metadata member is not valid. Content is required for the *attribute_name* attribute in the *element_name* element, but content was not found.

Explanation: The specified attribute is missing required content.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI012S   The XML structure of the *member_name* DB2 parameter metadata member is not valid. The content length for the *element_name* element cannot exceed *maximum_number* characters.

Explanation: The specified element contains too many characters.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI013S   The XML structure of the *member_name* DB2 parameter metadata member is not valid. The *attribute_name* attribute in the *element_name* element is unknown.

Explanation: The specified attribute in the DB2 parameter metadata member is unknown.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI014S   The content of the *member_name* DB2 parameter metadata member is not valid because the value of the *element_name* element is incorrect. The value is *value_name*.

Explanation: The specified value of the element is not a valid value.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI015S   The content of the DB2 parameter metadata member is not valid because the value of the *attribute_name* attribute in the *element_name* element is incorrect. The value of the attribute is *value_name*.

Explanation: The specified value of the attribute is not a valid value.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI016S   The content of the DB2 parameter metadata member is not valid because the data type of the *element_name* element is incorrect. The value is *value_name*.

Explanation: The specified data type is not a valid data type.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI017S   **The content of the DB2 parameter metadata member is not valid because the data type of the *attribute_name* attribute in the *element_name* element is incorrect. The value of the attribute is *value_name*.**

Explanation: The specified data type is not a valid data type.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI050S   **The *member_name* DB2 parameter metadata member was not found in the *data_set_name* data set.**

Explanation: Tools Customizer could not find the specified DB2 parameter metadata member.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI051S   **The *parameter_name* LPAR parameter in the *template_name* template does not have associated metadata in the *member_name* LPAR parameter metadata member.**

Explanation: The specified template does not contain metadata for an LPAR parameter. The name of the LPAR parameter metadata member, the name of the LPAR parameter, and the name of the template are indicated in the message text.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI052S   **The *parameter_name* product parameter in the *template_name* template does not have associated metadata in the *member_name* product parameter metadata member.**

Explanation: The specified template does not contain metadata for a product parameter. The name of the product parameter metadata member, the name of the product parameter, and the name of the template are indicated in the message text.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI053E   **The following metadata data set was not found: *data_set_name*.**

Explanation: Tools Customizer could not find the specified metadata data set.

System action: Processing stops.

User response: Ensure that the metadata data set is specified correctly. If the problem persists, contact IBM Software Support.

CCQI054E   **The following metadata data set could not be opened: *data_set_name*.**

Explanation: Tools Customizer could not open the specified LPAR metadata data set.

System action: Processing stops.

User response: Ensure the metadata data set was specified correctly.

CCQI055S   **The CCQ$$DB2 DB2 parameter metadata member was not found in the *data_set_name* Tools Customizer metadata data set.**

Explanation: Tools Customizer could not find the DB2 parameter metadata member in the specified Tools Customizer metadata data set.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI056S   **The CCQ$$LPR LPAR parameter metadata member was not found in the *data_set_name* data set.**

Explanation: Tools Customizer could not find the specified LPAR parameter metadata member.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI057S   **The *member_name* product parameter metadata member was not found in the *data_set_name* data set.**

Explanation: The product parameter metadata member was not found in the specified data set.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI058I *Product_name* **does not have any DB2 parameters.**

Explanation: DB2 parameters are not required to customize the specified product.

System action: Processing continues.

User response: No action is required.

CCQI059I *Product_name* **does not have any LPAR parameters.**

Explanation: LPAR parameters are not required to customize the specified product.

System action: Processing continues.

User response: No action is required.

CCQI060S **The** *parameter_name* **DB2 parameter in the** *task_description* **task condition does not have associated metadata in the** *member_name* **DB2 parameter metadata member.**

Explanation: Associated metadata is missing for the specified DB2 parameter in a task.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI061S **The** *parameter_name* **LPAR parameter in the** *task_description* **task condition does not have associated metadata in the** *member_name* **LPAR parameter metadata member.**

Explanation: Associated metadata is missing for the specified LPAR parameter in a task.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI062S **The** *parameter_name* **product parameter in the** *task_description* **task condition does not have associated metadata in the** *member_name* **product parameter metadata member.**

Explanation: Associated metadata is missing for the specified product parameter in a task.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI063S **The** *parameter_name* **DB2 parameter in the** *task_description* **task and the** *step_description* **step does not have associated metadata in the** *member_name* **DB2 parameter metadata member.**

Explanation: Associated metadata is missing for the specified DB2 parameter in a task and step.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI064S **The** *parameter_name* **LPAR parameter in the** *task_description* **task and the** *step_description* **step does not have associated metadata in the** *member_name* **LPAR parameter metadata member.**

Explanation: Associated metadata is missing for the specified LPAR parameter in a task and step.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI065S **The** *parameter_name* **product parameter in the** *task_description* **task and the** *step_description* **step does not have associated metadata in the** *member_name* **parameter metadata member.**

Explanation: Associated metadata is missing for the specified parameter in a task and step.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI066S **The** *parameter_name* **DB2 parameter in the** *task_description* **task,** *step_description* **step, and** *template_name* **template condition does not have associated metadata in the** *member_name* **DB2 parameter metadata member.**

Explanation: Associated metadata is missing for the specified DB2 parameter in a task, step, and template.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI067S The *parameter_name* **LPAR parameter in the** *task_description* **task,** *step_description* **step, and** *template_name* **template condition does not have associated metadata in the** *member_name* **LPAR parameter metadata member.**

Explanation: Associated metadata is missing for the specified LPAR parameter in a task, step, and template.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI068S The *parameter_name* **product parameter in the** *task_description* **task,** *step_description* **step, and** *template_name* **template condition does not have associated metadata in the** *member_name* **product parameter metadata member.**

Explanation: Associated metadata is missing for the specified product parameter in a task, step, and template.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI069S **Product metadata does not support multiple configurations, but the** *template_name* **product template contains the** *parameter_name* **parameter. Enable multiple configurations support for this product, and try again.**

Explanation: The specified template contains a parameter for multiple configurations, but the product is not enabled to support multiple configurations.

System action: Processing stops.

User response: Enable multiple configurations support, and try again.

CCQI070E The *parameter_name* **DB2 parameter metadata member is not valid. The default length for the** *parameter-element_name* **parameter element exceeds the length of the parameter. The default length is** *default_length***, and the specified length is** *specified_length***. The default length will be truncated accordingly.**

Explanation: The specified length cannot be shorter than the default length.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI071E The *parameter_name* **LPAR parameter metadata member is not valid. The default length for the** *parameter-element_name* **parameter element exceeds the length of the parameter. The default length is** *default_length***, and the specified length is** *specified_length***. The default length will be truncated accordingly.**

Explanation: The specified length cannot be shorter than the default length.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI072E The *parameter_name* **product parameter metadata member is not valid. The default length for the** *parameter-element_name* **parameter element exceeds the length of the parameter. The default length is** *default_length***, and the specified length is** *specified_length***. The default length will be truncated accordingly.**

Explanation: The specified length cannot be shorter than the default length.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI073S **The XML structure of the** *member_name* **DB2 parameter metadata member is not valid. The following value of the** *attribute_name* **attribute in the** *element_name* **element already exists:** *value_name***.**

Explanation: The specified value already exists for an attribute.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI074S The XML structure of the *member_name* LPAR parameter metadata member is not valid. The following value of the *attribute_name* attribute in the *element_name* element already exists: *value_name*.

Explanation: The specified value already exists for an attribute.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI075S The XML structure of the *member_name* product parameter metadata member is not valid. The following value of the *attribute_name* attribute in the *element_name* element already exists: *value_name*.

Explanation: The specified value already exists for an attribute.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI076S The XML structure of the *member_name* DB2 parameter metadata member is not valid. The *parameter_name* parameter refers to the *section-name* section. This section was not found in the DB2 parameter metadata member.

Explanation: The specified value already exists for an attribute.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI077S The XML structure of the *member_name* LPAR parameter metadata member is not valid. The *parameter_name* parameter refers to the *section-name* section. This section was not found in the LPAR parameter metadata member.

Explanation: The specified parameter refers to a section that is not in the LPAR parameter metadata member.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI078S The XML structure of the *member_name* product parameter metadata member is not valid. The *parameter_name* parameter refers to the *section-name* section. This section was not found in the product parameter metadata member.

Explanation: The specified parameter refers to a section that is not in the product parameter metadata member.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI080S The content of the *member_name* DB2 parameter metadata member is not valid because the value of the *attribute_name* attribute in the *element_name* element is incorrect. The value of the attribute is *value_name*.

Explanation: The specified value for an attribute in the DB2 parameter metadata member is not valid.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI081S The content of the *member_name* LPAR parameter metadata member is not valid because the value of the *attribute_name* attribute in the *element_name* element is incorrect. The value of the attribute is *value_name*.

Explanation: The specified value for an attribute in the LPAR parameter metadata member is not valid.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI082S The content of the *member_name* product parameter metadata member is not valid because the value of the *attribute_name* attribute in the *element_name* element is incorrect. The value of the attribute is *value_name*.

Explanation: The specified value for an attribute in the product parameter metadata member is not valid.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI090S **The product-defined DB2 parameter** *parameter_name* **in the** *member_name* **parameter metadata member references the** *section_ID* **section ID, but this ID does not exist in either the parameter metadata member or the DB2 parameter metadata member.**

Explanation: A section that does not exist in the parameter metadata member or the DB2 parameter metadata member is referenced by the specified DB2 parameter.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI091S **The product-defined LPAR parameter in the** *member_name* **parameter metadata member references the** *section_ID* **section ID, but this ID does not exist in either the parameter metadata member or the LPAR parameter metadata member.**

Explanation: A section that does not exist in the parameter metadata member or the LPAR parameter metadata member is being referenced by the specified LPAR parameter.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI092S **The overridden DB2 parameter** *parameter_name* **in the** *member_name* **parameter metadata member does not exist in the DB2 parameter metadata member.**

Explanation: The specified parameter does not exist.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI093S **The overridden LPAR parameter** *parameter_name* **in the** *member_name* **parameter metadata member does not exist in the LPAR parameter metadata member.**

Explanation: The specified parameter does not exist.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI094S **The CCQ$$PRD product customization parameter metadata member was not found in the** *data_set_name* **data set.**

Explanation: The specified data set must contain the CCQ$$PRD product customization parameter metadata member

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI100W **The XML structure of the** *member_name* **LPAR parameter metadata member is not valid. The PL/I XML parser issued the following exception warning code:** *code_number*.

Explanation: While determining if the LPAR parameter metadata member is valid, the PL/I XML parser issued an exception warning code.

System action: Processing continues.

User response: See the *Enterprise PL/I for z/OS Programming Guide* for more information about the exception warning code.

CCQI101S **The XML structure of the** *member_name* **LPAR parameter metadata member is not valid. The PL/I XML parser issued the following exception error code:** *code_number*.

Explanation: While determining if the LPAR parameter metadata member is valid, the PL/I XML parser issued an exception error code.

System action: Processing stops.

User response: See the *Enterprise PL/I for z/OS Programming Guide* for more information about the exception warning code.

CCQI102S **The XML structure of the** *member_name* **LPAR parameter metadata member is not valid. The** *element_name* **element is unknown.**

Explanation: The specified element in the LPAR parameter metadata member is unknown.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI103S The XML structure of the *member_name* LPAR parameter metadata member is not valid. Content is not allowed for the *element_name* element, but content was found.

Explanation: The specified element cannot contain content.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI104S The XML structure of the *member_name* LPAR parameter metadata member is not valid. Content is required for the *element_name* element, but content was not found.

Explanation: The specified element requires content.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI105S The XML structure of the *member_name* LPAR parameter metadata member is not valid. The content length for the *element_name* element cannot exceed *maximum_number* characters.

Explanation: The specified element contains too many characters.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI106S The XML structure of the *member_name* LPAR parameter metadata member is not valid. The content length for the *element_name* element must be at least *minimum_number* characters.

Explanation: The specified element does not contain enough characters.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI107S The XML structure of the *member_name* LPAR parameter metadata member is not valid. The *element_name* element must occur at least *minimum_number* times.

Explanation: The specified element does not occur enough times.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI108S The XML structure of the *member_name* LPAR parameter metadata member is not valid. The *attribute_name* attribute in the *element_name* element cannot occur more than *maximum_number* times.

Explanation: The specified attribute occurs too many times.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI109S The XML structure of the *member_name* LPAR parameter metadata member is not valid. The *attribute_name* attribute in the *element_name* element must occur at least *minimum_number* times.

Explanation: The specified attribute did not occur enough times.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI110S The XML structure of the *member_name* LPAR parameter metadata member is not valid. Content is not allowed for the *attribute_name* attribute in the *element_name* element, but content was found.

Explanation: The specified attribute cannot have content.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI111S **The XML structure of the** *member_name* **LPAR parameter metadata member is not valid. Content is required for the** *attribute_name* **attribute in the** *element_name* **element, but content was not found.**

Explanation: The specified attribute is missing required content.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI112S **The XML structure of the** *member_name* **LPAR parameter metadata member is not valid. The content length for the** *element_name* **element cannot exceed** *maximum_number* **characters.**

Explanation: The specified element contains too many characters.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI113S **The XML structure of the** *member_name* **LPAR parameter metadata member is not valid. The** *attribute_name* **attribute in the** *element_name* **element is unknown.**

Explanation: The specified attribute in the LPAR parameter metadata member is unknown.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI114S **The content of the** *member_name* **LPAR parameter metadata member is not valid because the value of the** *element_name* **element is incorrect. The value is** *value_name*.

Explanation: The specified value for an element in the LPAR parameter metadata member is not valid.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI115S **The content of the** *member_name* **LPAR parameter metadata member is not valid because the value of the** *attribute_name* **attribute in the** *element_name* **element is incorrect. The value of the attribute is** *value_name*.

Explanation: The specified value for an attribute in the LPAR parameter metadata member is not valid.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI116S **The content of the** *member_name* **LPAR parameter metadata member is not valid because the data type of the** *element_name* **element is incorrect. The value is** *value_name*.

Explanation: The specified data type value for an element in the LPAR parameter metadata member is not valid.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI117S **The content of the** *member_name* **LPAR parameter metadata member is not valid because the data type of the** *attribute_name* **attribute in the** *element_name* **element is incorrect. The value is** *value_name*.

Explanation: The specified data type value for an attribute in the LPAR parameter metadata member is not valid.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI120S **The XML structure of the** *member_name* **DB2 parameter metadata member is not valid. The** *element_name* **element in the** *parameter_name* **parameter contains duplicate values for the** *element_name* **element. The duplicate value is** *value_name*.

Explanation: An element contains the specified duplicate value.

System action: Processing stops.

User response: See "Gathering diagnostic

information" on page 902. Contact IBM Software
Support.

CCQI121S **The XML structure of the** *member_name*
**LPAR parameter metadata member is
not valid. The** *element_name* **element in
the** *parameter_name* **parameter contains
duplicate values for the** *element_name*
element. The duplicate value is
*value_name***.**

Explanation: An element contains the specified
duplicate value.

System action: Processing stops.

User response: See "Gathering diagnostic
information" on page 902. Contact IBM Software
Support.

CCQI122S **The XML structure of the** *member_name*
**parameter metadata member is not
valid. The** *element_name* **element in the**
parameter_name **parameter contains
duplicate values for the** *element_name*
element. The duplicate value is
*value_name***.**

Explanation: An element contains the specified
duplicate value.

System action: Processing stops.

User response: See "Gathering diagnostic
information" on page 902. Contact IBM Software
Support.

CCQI123S **The XML structure of the** *member_name*
**discover metadata member is not valid.
The** *element_name* **element in the**
parameter_name **parameter contains
duplicate values for the** *element_name*
element. The duplicate value is
*value_name***.**

Explanation: An element contains the specified
duplicate value.

System action: Processing stops.

User response: See "Gathering diagnostic
information" on page 902. Contact IBM Software
Support.

CCQI124S **The XML structure of the** *member_name*
**product customization parameter
metadata member is not valid. The**
element_name **element in the**
parameter_name **parameter contains
duplicate values for the** *element_name*
element. The duplicate value is
*value_name***.**

Explanation: An element contains the specified
duplicate value.

System action: Processing stops.

User response: See "Gathering diagnostic
information" on page 902. Contact IBM Software
Support.

CCQI200W **The XML structure of the** *member_name*
**information metadata member is not
valid. The PL/I XML parser issued the
following exception warning code:**
*code_number***.**

Explanation: While determining if the information
metadata member is valid, the PL/I XML parser issued
an exception warning code.

System action: Processing continues.

User response: See the *Enterprise PL/I for z/OS
Programming Guide* for more information about the
exception warning code.

CCQI201S **The XML structure of the** *member_name*
**information metadata member is not
valid. The PL/I XML parser issued the
following exception error code:**
*code_number***.**

Explanation: While determining if the information
metadata member is valid, the PL/I XML parser issued
an exception error code.

System action: Processing stops.

User response: See the *Enterprise PL/I for z/OS
Programming Guide* for more information about the
exception warning code.

CCQI202S **The XML structure of the** *member_name*
**information metadata member is not
valid. The** *element name* **element is
unknown.**

Explanation: The specified element in the information
metadata member is unknown.

System action: Processing stops.

User response: See "Gathering diagnostic
information" on page 902. Contact IBM Software
Support.

CCQI203S **The XML structure of the** *member_name*
**information metadata member is not
valid. Content is not allowed for the**
element_name **element, but content was
found.**

Explanation: The specified element cannot contain
content.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI204S **The XML structure of the** *member_name* **information metadata member is not valid. Content is required for the** *element_name* **element, but content was not found.**

Explanation: The specified element requires content.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI205S **The XML structure of the** *member_name* **information metadata member is not valid. The content length for the** *element_name* **element cannot exceed** *maximum_number* **characters.**

Explanation: The specified element contains too many characters.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI206S **The XML structure of the** *member_name* **information metadata member is not valid. The content length for the** *element_name* **element must be at least** *minimum_number* **characters.**

Explanation: The specified element does not contain enough characters.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI207S **The XML structure of the** *member_name* **information metadata member is not valid. The** *element_name* **element must occur at least** *minimum_number* **times.**

Explanation: The specified element does not occur enough times.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI208S **The XML structure of the** *member_name* **information metadata member is not valid. The** *attribute_name* **attribute in the** *element_name* **element cannot occur more than** *maximum_number* **times.**

Explanation: The specified attribute occurs too many times.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI209S **The XML structure of the** *member_name* **information metadata member is not valid. The** *attribute_name* **attribute in the** *element_name* **element must occur at least** *minimum_number* **times.**

Explanation: The specified attribute did not occur enough times.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI210S **The XML structure of the** *member_name* **information metadata member is not valid. Content is not allowed for the** *attribute_name* **attribute in the** *element_name* **element, but content was found.**

Explanation: The specified attribute cannot have content.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI211S **The XML structure of the** *member_name* **information metadata member is not valid. Content is required for the** *attribute_name* **attribute in the** *element_name* **element, but content was not found.**

Explanation: The specified attribute is missing required content.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI212S **The XML structure of the** *member_name* **information metadata member is not valid. The content length for the** *element_name* **element cannot exceed** *maximum_number* **characters.**

Explanation: The specified element contains too many characters.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI213S **The XML structure of the** *member_name* **information metadata member is not valid. The** *attribute_name* **attribute in the** *element_name* **element is unknown.**

Explanation: The specified attribute in the information metadata member is unknown.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI214S **The content of the** *member_name* **information metadata member is not valid because the value of the** *element_name* **element is incorrect. The value is** *value_name***.**

Explanation: The specified value for an element in the information metadata member is not valid.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI215S **The content of the** *member_name* **information metadata member is not valid because the value of the** *attribute_name* **attribute in the** *element_name* **element is incorrect. The value is** *value_name***.**

Explanation: The specified value for an attribute in the information metadata member is not valid.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI216S **The content of the** *member_name* **information metadata member is not valid because the data type of the** *element_name* **element is incorrect. The value is** *value_name***.**

Explanation: The specified data type value for an element in the information metadata member is not valid.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI217S **The content of the** *member_name* **information metadata member is not valid because the data type of the** *attribute_name* **attribute in the** *element_name* **element is incorrect. The value is** *value_name***.**

Explanation: The specified data type value for an attribute in the information metadata member is not valid.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI218S **The content of the** *member_name* **information metadata member is not valid. The length of the** *value_name* **value that of the** *attribute_name* **attribute is longer than the** *value_name* **value of the** *attribute_name* **attribute.**

Explanation: The first specified value cannot be longer than the second specified value.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI219S **The content of the** *member_name* **information metadata member is not valid. The** *value_name* **value of the** *attribute_name* **attribute contains the** *value_name* **value.**

Explanation: The first specified value cannot be longer than the second specified value.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI220S   The XML structure of the *member_name* **information metadata member is not valid. Content for the** *attribute_name* **attribute in the***element_name* **element exceed** *maximum_number* **characters.**

Explanation: The specified attribute contains too many characters.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI223S   The XML structure of the *member_name* **information metadata member is not valid. The value that is specified for the DB2 Level already exists. The value is** *value_name***.**

Explanation: The specified value already exists.

System action: Processing stops.

User response: Specify a different DB2 level. If the problem persists, contact IBM Software Support.

CCQI224S   The XML structure of the *member_name* **information metadata member is not valid. The value that is specified for the DB2 Mode already exists. The value is** *value_name***.**

Explanation: The specified value already exists.

System action: Processing stops.

User response: Specify a different DB2 mode. If the problem persists, contact IBM Software Support.

CCQI250S   The information metadata member was not found in the *data_set_name* **data set.**

Explanation: Tools Customizer could not find the information metadata member in the specified data set.

System action: Processing stops.

User response: If this message was issued on the Specify the Metadata Library (CCQPHLQ) panel, specify the product metadata library. The name of this library is *hlq*.SADBDENU.

Do not specify the Tools Customizer metadata library, which is *hlq*.SCCQDENU.

If the problem persists, identify the name of the Tools Customizer trace data set and contact IBM Software Support.

CCQI251E   The *member_name* **member was not accessible in the** *data_set_name* **data set.**

Explanation: The specified member could not be accessed in the data set.

System action: Processing stops.

User response: Specify the correct metadata library.

CCQI252S   The information metadata member was not found in the *library_name* **component metadata library that is part of the** *library_name* **pack metadata library. The name of the pack is** *pack_name***.**

Explanation: The specified component metadata library does not contain the information metadata member.

System action: Processing stops.

User response: Specify the correct metadata library.

CCQI253E   The *library_name* **Tools Customizer metadata library is not current. Update the metadata library on the Tools Customizer Settings panel.**

Explanation: The specified metadata library is not current.

System action: Processing stops.

User response: Specify a current metadata library on the Tools Customizer Settings panel.

CCQI300W   The XML structure of the *member_name* **sequence metadata member is not valid. The PL/I XML parser issued the following exception warning code:** *code_number***.**

Explanation: While determining if the sequence metadata member is valid, the PL/I XML parser issued an exception warning code.

System action: Processing continues.

User response: See the *Enterprise PL/I for z/OS Programming Guide* for more information about the exception warning code.

CCQI301S   The XML structure of the *member_name* **sequence metadata member is not valid. The PL/I XML parser issued the following exception error code:** *code_number***.**

Explanation: While determining if the sequence metadata member is valid, the PL/I XML parser issued an exception error code.

System action: Processing stops.

User response: See the *Enterprise PL/I for z/OS Programming Guide* for more information about the exception error code, and contact IBM Software Support.

CCQI302S **The XML structure of the** *member_name* **sequence metadata member is not valid. The** *element_name* **element is unknown.**

Explanation: The specified element in the sequence metadata member is unknown.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI303S **The XML structure of the** *member_name* **sequence metadata member is not valid. Content is not allowed for the** *element_name* **element, but content was found.**

Explanation: The specified element cannot contain content.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI304S **The XML structure of the** *member_name* **sequence metadata member is not valid. Content is required for the** *element_name* **element, but content was not found.**

Explanation: The specified element is missing required content.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI305S **The XML structure of the** *member_name* **sequence metadata member is not valid. Content length for the** *element_name* **element cannot exceed** *maximum_number* **characters.**

Explanation: The specified element contains too many characters.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI306S **The XML structure of the** *member_name* **sequence metadata member is not valid. The** *element_name* **element cannot occur more than** *maximum_number* **times.**

Explanation: The specified element occurs too many times.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI307S **The XML structure of the** *member_name* **sequence metadata member is not valid. The** *element_name* **element must occur at least** *minimum_number* **times.**

Explanation: The specified element does not occur enough times.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI308S **The XML structure of the** *member_name* **sequence metadata member is not valid. The** *attribute_name* **attribute in the** *element_name* **element cannot occur more than** *maximum_number* **times.**

Explanation: The specified attribute occurs too many times.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI309S **The XML structure of the** *member_name* **sequence metadata member is not valid. The** *attribute_name* **attribute in the** *element_name* **element must occur at least** *minimum_number* **times.**

Explanation: The specified attribute does not occur enough times.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI310S **The XML structure of the** *member_name* **sequence metadata member is not valid. Content is not allowed for the** *attribute_name* **attribute in the** *element_name* **element, but content was found.**

Explanation: The specified attribute cannot contain content.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI311S **The XML structure of the** *member_name* **sequence metadata member is not valid. Content is required for the** *attribute_name* **attribute in the** *element_name* **element, but content was not found.**

Explanation: The specified attribute is missing required content.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI312S **The XML structure of the** *member_name* **sequence metadata member is not valid. The content length for the** *element_name* **element cannot exceed** *maximum_number* **characters.**

Explanation: The specified element contains too many characters.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI313S **The XML structure of the** *member_name* **sequence metadata member is not valid. The** *attribute_name* **attribute in the** *element_name* **element is unknown.**

Explanation: The specified attribute in the sequence metadata member is unknown.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI314S **The content of the** *member_name* **sequence metadata member is not valid because the value of the** *element_name* **element is incorrect. The value is** *value_name*.

Explanation: The specified value for an element in the sequence metadata member is not valid.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI315S **The content of the** *member_name* **sequence metadata member is not valid because the value of the** *attribute_name* **attribute in the** *element_name* **element is incorrect. The value is** *value_name*.

Explanation: The specified value for an attribute in the sequence metadata member is not valid.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI316S **The content of the** *member_name* **sequence metadata member is not valid because the data type of the** *element_name* **element is incorrect. The value is** *value_name*.

Explanation: The specified data type value for an element in the sequence metadata member is not valid.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI317S **The content of the** *member_name* **sequence metadata member is not valid because the data type of the** *attribute_name* **attribute in the** *element_name* **element is incorrect. The value is** *value_name*.

Explanation: The specified data type value for an attribute in the sequence metadata member is not valid.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI350S **The XML structure of the** *member_name* **sequence metadata member is not valid because the value of the** *attribute_name* **attribute in the** *element_name* **element is incorrect. The value is** *value_name*.

Explanation: A specified value for an attribute in the sequence metadata member is not valid.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI351S The *member_name* **sequence metadata member was not found in the** *data_set_name* **metadata data set.**

Explanation: Tools Customizer could not find the specified sequence metadata member in the metadata data set.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI352S The *template_name* **product template was not found in the** *data_set_name* **metadata data set.**

Explanation: Tools Customizer could not find the specified product template in the data set.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI353S **The sequence metadata member was not found in the** *data_set_name* **component data set that is part of the** *data_set_name* **pack.**

Explanation: Tools Customizer could not find the sequence metadata member.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI360S The XML structure of the *member_name* **sequence metadata member is not valid. The value of the** *attribute_name* **attribute in the** *element_name* **element already exists.**

Explanation: The specified attribute contains a value that already exists.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI361S The XML structure of the *member_name* **sequence metadata member is not valid. The condition element on the** *level_type* **level already contains a relational operator.**

Explanation: A relational operator already exists for the condition element on the specified level.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI362S The XML structure of the *member_name* **sequence metadata member is not valid. The condition element on the** *level_type* **level must contain only one content string or content number element.**

Explanation: Only one content string element or content number element can be contained in the condition element on the specified level.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI363S The XML structure of the *member_name* **sequence metadata member is not valid. The condition element in the** *element_name* **element with the** *attribute_name* **attribute must contain either the content string element or content number element.**

Explanation: Either the content string element or the content number element must be in the condition element.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI400W The XML structure of the *member_name* **parameter metadata member is not valid. The PL/I XML parser issued the following exception warning code:** *code_number***.**

Explanation: While determining the parameter metadata member is valid, the PL/I XML parser issued an exception warning code.

System action: Processing continues.

User response: See the *Enterprise PL/I for z/OS Programming Guide* for more information about the exception warning code.

CCQI401S The XML structure of the *member_name* **parameter metadata member is not valid. The PL/I XML parser issued the following exception error code:** *code_number***.**

Explanation: While determining if the parameter

metadata member is valid, the PL/I XML parser issued an exception error code.

System action: Processing stops.

User response: See the *Enterprise PL/I for z/OS Programming Guide* for more information about the exception warning code.

CCQI402S **The XML structure of the** *member_name* **parameter metadata member is not valid. The** *element name* **element is unknown.**

Explanation: The specified element in the parameter metadata member is unknown.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI403S **The XML structure of the** *member_name* **parameter metadata member is not valid. Content is not allowed for the** *element_name* **element, but content was found.**

Explanation: The specified element cannot contain content.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI404S **The XML structure of the** *member_name* **parameter metadata member is not valid. Content is required for the** *element_name* **element, but content was not found.**

Explanation: The specified element requires content.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI405S **The XML structure of the** *member_name* **parameter metadata member is not valid. The content length for the** *element_name* **element cannot exceed** *maximum_number* **characters.**

Explanation: The specified element contains too many characters.

System action: Processing stops.

User response: See "Gathering diagnostic

information" on page 902. Contact IBM Software Support.

CCQI406S **The XML structure of the** *member_name* **parameter metadata member is not valid. The content length for the** *element_name* **element must be at least** *minimum_number* **characters.**

Explanation: The specified element does not contain enough characters.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI407S **The XML structure of the** *member_name* **parameter metadata member is not valid. The** *element_name* **element must occur at least** *minimum_number* **times.**

Explanation: The specified element does not occur enough times.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI408S **The XML structure of the** *member_name* **parameter metadata member is not valid. The** *attribute_name* **attribute in the** *element_name* **element cannot occur more than** *maximum_number* **times.**

Explanation: The specified attribute occurs too many times.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI409S **The XML structure of the** *member_name* **parameter metadata member is not valid. The** *attribute_name* **attribute in the** *element_name* **element must occur at least** *minimum_number* **times.**

Explanation: The specified attribute does not occur enough times.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI410S The XML structure of the *member_name* parameter metadata member is not valid. Content is not allowed for the *attribute_name* attribute in the *element_name* element, but content was found.

Explanation: The specified attribute cannot have content.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI411S The XML structure of the *member_name* parameter metadata member is not valid. Content is required for the *attribute_name* attribute in the *element_name* element, but content was not found.

Explanation: The specified attribute is missing required content.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI412S The XML structure of the *member_name* parameter metadata member is not valid. The content length for the *element_name* element cannot exceed *maximum_number* characters.

Explanation: The specified element contains too many characters.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI413S The XML structure of the *member_name* parameter metadata member is not valid. The *attribute_name* attribute in the *element_name* element is unknown.

Explanation: The specified attribute in the parameter metadata member is unknown.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI414S The content of the *member_name* parameter metadata member is not valid because the value of the *element_name* element is incorrect. The value is *value_name*.

Explanation: The specified value for an element in the parameter metadata member is not valid.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI415S The content of the *member_name* parameter metadata member is not valid because the value of the *attribute_name* attribute in the *element_name* element is incorrect. The value is *value_name*.

Explanation: The specified value for an attribute in the parameter metadata member is not valid.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI416S The content of the *member_name* parameter metadata member is not valid because the data type of the *element_name* element is incorrect. The value is *value_name*.

Explanation: The specified data type value for an element in the parameter metadata member is not valid.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI417S The content of the *member_name* parameter metadata member is not valid because the data type of the *attribute_name* attribute in the *element_name* element is incorrect. The value is *value_name*.

Explanation: The specified data type value for an attribute in the parameter metadata member is not valid.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI420S   The XML structure of the *member_name* **parameter metadata member is not valid. The** *element_name* **element is unknown for the overridden DB2 parameter.**

Explanation:

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI421S   The XML structure of the *member_name* **parameter metadata member is not valid. The** *element_name* **element is unknown for the overridden LPAR parameter.**

Explanation:

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI422S   The XML structure of the *member_name* **parameter metadata member is not valid. The** *attribute_name* **attribute in the** *element_name* **element is unknown for the overridden DB2 parameter.**

Explanation:

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI423S   The XML structure of the *member_name* **parameter metadata member is not valid. The** *attribute_name* **attribute in the** *element_name* **element is unknown for the overridden LPAR parameter.**

Explanation:

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI450S   The *member_name* **product parameter metadata member was not found in the** *data_set_name* **data set.**

Explanation: Tools Customizer could not find the specified product parameter metadata member.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI510W   The *data_set_name* **data store data set does not exist.**

Explanation: The specified data store data set does not exist.

System action: Processing continues.

User response: Ensure that the data store data set exists.

CCQI511S   The *data_set_name* **data store data set cannot be opened by using the** *disposition_type* **disposition.**

Explanation: The specified data store data set could not be opened with the specified disposition.

System action: Processing continues.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI512S   The *data_set_name* **data store data set cannot be opened by using the** *option-type* **option.**

Explanation: The specified data store data set was unable to be opened with the specified option.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI600W   The XML structure of the *member_name* **product customization parameter metadata member is not valid. The PL/I XML parser issued the following exception warning code:** *code_number***.**

Explanation: While determining if the product customization parameter metadata member is valid, the PL/I XML parser issued an exception warning code.

System action: Processing continues.

User response: See the *Enterprise PL/I for z/OS Programming Guide* for more information about the warning.

CCQI601S   The XML structure of the *member_name* **product customization parameter metadata member is not valid. The PL/I XML parser issued the following exception error code:** *code_number***.**

Explanation: While determining if the product

customization parameter metadata member is valid, the PL/I XML parser issued an exception error code.

System action: Processing continues.

User response: See the *Enterprise PL/I for z/OS Programming Guide* for more information about the warning.

CCQI602S   The XML structure of the *member_name* **product customization parameter metadata member is not valid. The** *element_name* **element is unknown.**

Explanation: The specified product customization parameter metadata member contains an unknown element.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI603S   The XML structure of the *member_name* **product customization parameter metadata member is not valid. Content is not allowed for the** *element_name* **element, but content was found.**

Explanation: Content was found in an element that cannot contain content.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI604S   The XML structure of the *member_name* **product customization parameter metadata member is not valid. Content is required for the** *element_name* **element, but content was not found.**

Explanation: The specified element does not contain required content.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI605S   The XML structure of the *member_name* **product customization parameter metadata member is not valid. The content length for the** *element_name* **element 'cannot exceed** *maximum_number* **characters.**

Explanation: The specified element contains too many characters.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI606S   The XML structure of the *member_name* **product customization parameter metadata member is not valid. The** *element_name* **element cannot occur more than** *maximum_number* **times.**

Explanation: The specified element occurs too many times in the product customization parameter metadata member.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI607S   The XML structure of the *member_name* **product customization parameter metadata member is not valid. The** *element_name* **element must occur at least** *minimum_number* **times.**

Explanation: The specified element does not occur enough times in the product customization parameter metadata member.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI608S   The XML structure of the *member_name* **product customization parameter metadata member is not valid. The** *attribute_name* **attribute in the** *element_name* **element cannot occur more than** *maximum_number* **times.**

Explanation: The specified attribute occurs too many times in the product customization parameter metadata member.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI609S   The XML structure of the *member_name* **product customization parameter metadata member is not valid. The** *attribute_name* **attribute in the** *element_name* **element must occur at least** *minimum_number* **times.**

Explanation: The specified attribute does not occur enough times in the product customization parameter metadata member.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI610S The XML structure of the *member_name* product customization parameter metadata member is not valid. Content is not allowed for the *attribute_name* attribute in the *element_name* element, but content was found.

Explanation: Content was found in an element that cannot contain content.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI611S The XML structure of the *member_name* product customization parameter metadata member is not valid. Content is required for the *attribute_name* attribute 'in the *element_name* element, but content was not found.

Explanation: The specified attribute does not contain required content.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI612S The XML structure of the *member_name* product customization parameter metadata member is not valid. The content length for the *attribute_name* attribute in the *element_name* element cannot exceed *maximum_number* characters.

Explanation: The specified attribute contains too many characters.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI613S The XML structure of the *member_name* product customization parameter metadata member is not valid. The *attribute_name* attribute in the *element_name* element is unknown.

Explanation: The specified product customization parameter metadata member contains an unknown attribute.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI614S The XML structure of the *member_name* product customization parameter metadata member is not valid. The value of the *element_name* element is not valid. The value *value_name*.

Explanation: The specified value of the element is not a valid value.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI615S The XML structure of the *member_name* product customization parameter metadata member is not valid. The value of the *attribute_name* attribute for the *element_name* element is not valid. The value is *value_name*.

Explanation: The specified value of the attribute is not a valid value.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI616S The XML structure of the *member_name* product customization parameter metadata member is not valid. The data type of the *element_name* element is 'not valid. The value of the element is *value_name*.

Explanation: The specified data type is not a valid data type.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI617S The XML structure of the *member_name* product customization parameter metadata member is not valid. The data type of the *attribute_name* attribute for the *element_name* element is not valid. The value of the attribute is *value_name*..

Explanation: The specified data type is not a valid data type.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI650S **The XML structure of the** *member_name* **product customization parameter metadata member is not valid. The following value of the** *attribute_name* **attribute in the** *element_name* **element already exists:** *value_name***.**

Explanation: The specified value for an attribute already exists.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI651S **The XML structure of the** *member_name* **product customization parameter metadata member is not valid. The** *parameter_name* **parameter refers to the following section, which was not found in the** *member_name* **product customization parameter metadata member:** *section-name***.**

Explanation: The specified section is not in the product customization parameter metadata member.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI652S **The** *member_name* **product customization metadata member not valid. The default length for the** *element_name* **parameter element exceeds the length of the parameter. The default length is** *default_length***, and the specified length is** *specified_length***. The default length will be truncated accordingly.**

Explanation: The specified length cannot be shorter than the default length.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI653S **The content of the** *member_name* **product customization parameter metadata member is not valid. The value of the** *attribute_name* **attribute in the** *element_name* **element is not valid. The value of the attribute is** *value_name***.**

Explanation: The specified value of the attribute is not a valid value.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI700W **The XML structure of the** *member_name* **solution pack metadata member is not valid. The PL/I XML parser issued the following exception warning code:** *code_number***.**

Explanation: While determining if the specified solution pack metadata member is valid, the PL/I XML parser issued an exception warning code.

System action: Processing continues.

User response: See the *Enterprise PL/I for z/OS Programming Guide* for more information about the warning.

CCQI701S **The XML structure of the** *member_name* **solution pack metadata member is not valid. The PL/I XML parser issued the following exception error code:** *code_number***.**

Explanation: While determining if the specified solution pack metadata member is valid, the PL/I XML parser issued an exception error code.

System action: Processing stops.

User response: See the *Enterprise PL/I for z/OS Programming Guide* for more information about the error.

CCQI702S **The XML structure of the** *member_name* **solution pack metadata member is not valid. The** *element_name* **element is unknown.**

Explanation: The specified solution pack metadata member contains an unknown element.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI703S **The XML structure of the** *member_name* **solution pack metadata member is not valid. Content is not allowed for the** *element_name* **element, but content was found**

Explanation: Content was found in an element that cannot contain content.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI704S   **The XML structure of the** *member_name* **solution pack metadata member is not valid. Content is required for the** *element_name* **element, but content was not found.**

Explanation: The specified element does not contain required content.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI705S   **The XML structure of the** *member_name* **solution pack metadata member is not valid. The content length for the** *element_name* **element cannot exceed** *maximum_number* **characters.**

Explanation: The specified element contains too many characters.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI706S   **The XML structure of the** *member_name* **solution pack metadata member is not valid. The** *element_name* **element cannot occur more than** *maximum_number* **times.**

Explanation: The specified element occurs too many times.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI707S   **The XML structure of the** *member_name* **solution pack metadata member is not valid. The** *element_name* **element must occur at least** *minimum_number* **times.**

Explanation: The specified element does not occur enough times.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI708S   **The XML structure of the** *member_name* **solution pack metadata member is not valid. The** *attribute_name* **attribute in the** *element_name* **element cannot occur more than** *maximum_number* **times.**

Explanation: The specified attribute occurs too many times.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI709S   **The XML structure of the** *member_name* **solution pack metadata member is not valid. The** *attribute_name* **attribute in the** *element_name* **element must occur at least** *minimum_number* **times.**

Explanation: The specified attribute does not occur enough times.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI710S   **The XML structure of the** *member_name* **solution pack metadata member is not valid. Content is not allowed for the** *attribute_name* **attribute in the** *element_name* **element, but content was found.**

Explanation: The specified attribute cannot have content.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI711S   **The XML structure of the** *member_name* **solution pack metadata member is not valid. Content is required for the** *attribute_name* **attribute in the** *element_name* **element, but content was not found.**

Explanation: The specified attribute is missing content.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI712S	**The XML structure of the** *member_name* **solution pack metadata member is not valid. The content length for the** *attribute_name* **attribute in the** *element_name* **element cannot exceed** *maximum_number* **characters.**

Explanation: The specified attribute contains too many characters.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI713S	**The XML structure of the** *member_name* **solution pack metadata member is not valid. The** *attribute_name* **attribute in the** *element_name* **element is unknown.**

Explanation: The specified attribute in the solution pack metadata member is unknown.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI714S	**The XML structure of the** *member_name* **solution pack metadata member is not valid because the value of the** *element_name* **element is incorrect. The value is** *value_name*.

Explanation: The specified value of the element is not a valid value.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI715S	**The XML structure of the** *member_name* **solution pack metadata member is not valid because the value of the** *attribute_name* **attribute in the** *element_name* **element is incorrect. The value of the attribute is** *value_name*.

Explanation: The specified value of the attribute is not a valid value.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI716S	**The XML structure of the** *member_name* **solution pack metadata member is not valid because the data type of the** *element_name* **element is incorrect. The value is** *value_name*.

Explanation: The specified data type is not a valid data type.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI717S	**The XML structure of the** *member_name* **solution pack metadata member is not valid because the data type of the** *attribute_name* **attribute in the** *element_name* **element is incorrect. The value of the attribute is** *value_name*.

Explanation: The specified data type is not a valid data type.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI720S	**The XML structure of the** *member_name* **solution pack metadata member is not valid. The msg element is required for the** *component_name* **component that is not customizable.**

Explanation: The msg element is required for the specified component, which cannot be customized by using Tools Customizer.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI750S	**The solution pack metadata member was not found in the** *library_name* **metadata library.**

Explanation: Tools Customizer could not find the solution pack metadata member in the specified library.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI751S   **The version in the** *library_name* **solution pack metadata library is different than the version in the** *library_name* **component metadata library. The name of the pack is** *pack_name***, and the name of the component is** *component_name***.**

Explanation: The version in the solution pack metadata library does not match the version in the component metadata library.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI752S   **The release in the** *library_name* **solution pack metadata library is different than the release in the** *library_name* **component metadata library. The name of the pack is** *pack_name***, and the name of the component is** *component_name***.**

Explanation: The release in the solution pack metadata library does not match the release in the component metadata library.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQI753S   **The modification level in the** *library_name* **solution pack metadata library is different than the modification level in the** *library_name* **component metadata library. The name of the pack is** *pack_name***, and the name of the component is** *component_name***.**

Explanation: The modification level in the solution pack metadata library does not match the modification level in the component metadata library.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQM002E   **The** *command_name* **line command is not valid: .**

Explanation: The specified line command is not valid.

System action: Processing continues.

User response: Specify a valid line command on the panel.

CCQO000W   **The XML structure of the** *member_name* **discover parameter metadata member is not valid. The PL/I XML parser issued the following exception warning code:** *code_number***.**

Explanation: While determining if the discover parameter metadata member is valid, the PL/I XML parser issued an exception warning code.

System action: Processing continues.

User response: See the *Enterprise PL/I for z/OS Programming Guide* for more information about the exception warning code.

CCQO001S   **The XML structure of the** *member_name* **discover parameter metadata member is not valid. The PL/I XML parser issued the following exception error code:** *code_number***.**

Explanation: While determining if the Discover metadata member is valid, the PL/I XML parser issued an exception error code.

System action: Processing stops.

User response: See the *Enterprise PL/I for z/OS Programming Guide* for more information about the exception warning code. Contact IBM Software Support.

CCQO002S   **The XML structure of the** *member_name* **discover parameter metadata member is not valid. The** *element_name* **element is unknown.**

Explanation: The specified element in the discover parameter metadata member is unknown.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQO003S   **The XML structure of the** *member_name* **discover parameter metadata member is not valid. Content is not allowed for the** *element_name* **element, but content was found.**

Explanation: The specified element cannot contain content.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQO004S The XML structure of the *member_name* discover parameter metadata member is not valid. Content is required for the *element name* **element, but content was not found.**

Explanation: The specified element is missing required content.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQO005S The XML structure of the *member_name* discover parameter metadata member is not valid. The content length for the *element_name* **element cannot exceed** *maximum_number* **characters.**

Explanation: The specified element contains too many characters.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQO006S The XML structure of the *member_name* discover parameter metadata member is not valid. The *element_name* **element cannot occur more than** *maximum_number* **times.**

Explanation: The specified element occurs too many times.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQO007S The XML structure of the *member_name* discover parameter metadata member is not valid. The *element_name* **element must occur at least** *minimum_number* **times.**

Explanation: The specified element does not occur enough times.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQO008S The XML structure of the *member_name* discover parameter metadata member is not valid. The *attribute_name* **attribute in the** *element_name* **element cannot occur more than** *maximum_number* **times.**

Explanation: The specified attribute occurs too many times.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQO009S The XML structure of the *member_name* discover parameter metadata member is not valid. The *attribute_name* **attribute in the** *element_name* **element must occur at least** *minimum_number* **times.**

Explanation: The specified attribute does not occur enough times.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQO010S The XML structure of the *member_name* discover parameter metadata member is not valid. Content is not allowed for the *attribute_name* **attribute in the** *element_name* **element, but content was found.**

Explanation: The specified attribute cannot contain content.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQO011S The XML structure of the *member_name* discover parameter metadata member is not valid. Content is required for the *attribute_name* **attribute in the** *element_name* **element, but content was not found.**

Explanation: The specified attribute requires content.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQO012S **The XML structure of the** *member_name*
discover parameter metadata member is
not valid. The content length for the
attribute_name **attribute in the**
element_name **element in the cannot**
exceed *maximum_number* **characters.**

Explanation: The specified attribute contains too many
characters.

System action: Processing stops.

User response: See "Gathering diagnostic
information" on page 902. Contact IBM Software
Support.

CCQO013S **The XML structure of the** *member_name*
discover parameter metadata member is
not valid. The *attribute_name* **attribute in**
the *element_name* **element is unknown.**

Explanation: The specified attribute is unknown.

System action: Processing stops.

User response: See "Gathering diagnostic
information" on page 902. Contact IBM Software
Support.

CCQO014S **The content of the** *member_name* **discover**
parameter metadata member is not valid
because the value of the *element_name*
element is incorrect. The value is
value_name.

Explanation: A The specified value for an element in
the discover parameter metadata member is not valid.

System action: Processing stops.

User response: See "Gathering diagnostic
information" on page 902. Contact IBM Software
Support.

CCQO015S **The content of the** *member_name* **discover**
parameter metadata member is not valid
because the value of the *attribute_name*
attribute in the *element_name* **element is**
incorrect. The value is *value_name*.

Explanation: The specified value for an attribute in
the discover parameter metadata member is not valid.

System action: Processing stops.

User response: See "Gathering diagnostic
information" on page 902. Contact IBM Software
Support.

CCQO016S **The content of the** *member_name* **discover**
parameter metadata member is not valid
because the data type of the
element_name **element is incorrect. The**
value is *value_name*.

Explanation: The specified data type value for an
element in the discover parameter metadata member is
not valid.

System action: Processing stops.

User response: See "Gathering diagnostic
information" on page 902. Contact IBM Software
Support.

CCQO017S **The content of the** *member_name* **product**
parameter metadata member is not valid
because the data type of the
attribute_name **attribute in the**
element_name **element is incorrect. The**
value is *value_name*.

Explanation: The specified data type value for an
attribute in the product parameter metadata member is
not valid.

System action: Processing stops.

User response: See "Gathering diagnostic
information" on page 902. Contact IBM Software
Support.

CCQO050S **The** *data_set_name* **Discover REXX EXEC**
data set could not be initialized or was
not found.

Explanation: Tools Customizer could not find or could
not initialize the specified Discover REXX EXEC data
set.

System action: Processing stops.

User response: Ensure that the Discover REXX EXEC
is specified correctly.

CCQO051W **The** *data_sharing_group_ID* **data sharing**
group ID cannot contain more than four
characters.

Explanation: The specified data sharing group ID
contains too many characters.

System action: Processing continues.

User response: Ensure that the specified data sharing
group ID does not exceed four characters.

CCQO052S **The** *REXX_EXEC_name* **Discover REXX**
EXEC was not found in the
data_set_name **Discover data set.**

Explanation: Tools Customizer could not find the
Discover REXX EXEC in the specified data set.

System action: Processing stops.

User response: Ensure that the Discover data set was specified correctly.

CCQO053W The *LPAR_name* **LPAR name cannot contain more than eight characters.**

Explanation: The specified LPAR name contains too many characters.

System action: Processing continues.

User response: Ensure that the specified LPAR name does not exceed eight characters.

CCQO054W The *subsystem_ID* **DB2 SSID cannot contain more than four characters. The record was not processed.**

Explanation: The specified DB2 SSID contains too many characters.

System action: Processing continues.

User response: Ensure that the specified DB2 SSID does not exceed four characters.

CCQO055W The *parameter_name* **DB2 group attach name parameter is in the** *record_name* **Discover record, but a DB2 group attach name was not specified. The record was not processed.**

Explanation: The Discover record contains a data sharing group parameter, but a DB2 group attach name was not specified.

System action: Processing continues.

User response: Ensure that information is specified correctly on the Discover Customized Product Information panel.

CCQO056W The *parameter_name* **DB2 parameter in the** *record_name* **Discover record did not have a DB2 group attach name or a DB2 SSID. The record was not processed.**

Explanation: The Discover record did not have a DB2 group attach name or a DB2 subsystem ID in the DB2 parameter.

System action: Processing continues.

User response: Ensure that information is specified correctly on the Discover Customized Product Information panel.

CCQO057W The Discover EXEC could not find the *parameter_name* **parameter in the metadata for the product to be customized. The record was not processed.**

Explanation: The specified parameter could not be found in the metadata for the product to be customized.

System action: Processing continues.

User response: Ensure that information is specified correctly on the Discover Customized Product Information panel.

CCQO058W The *parameter_name* **product parameter name in the** *record_type* **Discover record does not start with CCQ_LPR_, CCQ_DB2_, or CCQ_PRD_. The record was not processed.**

Explanation: The parameter in the record does not start with CCQ_DB2_, CCQ_LPR_, or CCQ_PRD_.

System action: Processing continues.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQO059W The *parameter_name* **product parameter cannot contain more than 72 characters. The record was not processed.**

Explanation: The specified product parameter contains too many characters.

System action: Processing continues.

User response: Ensure that the specified product parameter does not exceed 72 characters.

CCQO060W The *record_name* **Discover record from the REXX EXEC output must start with the following record type:** *record_type***. The record was not processed.**

Explanation: A Discover record from the REXX EXEC output must start with the specified DB2 record type.

System action: Processing continues.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQO061I If you do not have a previously customized version of the product, do not run the Discover EXEC. Press END to go to the Customizer Workplace panel.

Explanation: This message is issued when you customize a product for a the first time. It prompts you to use the Discover EXEC to discover data from a previous customization of the specified product.

System action: Processing continues.

User response:

Tip: Using the Discover EXEC saves time and reduces errors that can error when parameters are specified manually. If you want to use the Discover EXEC, specify the required information on the Discover Customized Product Information panel. Otherwise, press End to continue without discovering data from a previous customization of the product.

CCQO062W   The Discover EXEC could not find the following *parameter_name* parameter in the DB2 metadata. The record was not processed.

Explanation: The specified parameter is missing in the DB2 metadata.

System action: Processing continues.

User response: If this parameter is required, contact IBM Software Support.

CCQO064W   The *Discover-record* Discover record did not have a parameter name. The record was not processed.

Explanation: A parameter name was missing in the Discover record.

System action: Processing continues.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQO065W   The value for the *parameter_name* parameter is ignored because it has more than *maximum_number* characters, which is the maximum length that is defined in the metadata. The value is *parameter_value*.

Explanation: The specified value exceeded the maximum allowed length, which was defined in the metadata. Tools Customizer truncated the extra characters.

System action: Processing continues.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQO066W   The *record_name* Discover record from the Discover REXX EXEC output does not have a parameter value. The record was not processed.

Explanation: The Discover record was missing a parameter value from the Discover EXEC output.

System action: Processing continues.

User response: Ensure that information was specified

correctly on the Discover Customized Product Information panel.

CCQO067W   The *parameter_name* parameter is defined in the metadata to support one value, but more than one value was found. The last value was used.

Explanation: The definition of the parameter in the metadata supports one value, but more than one value was specified. Only the last value was used.

System action: Processing continues.

User response: Ensure that information was specified correctly on the Discover Customized Product Information panel.

CCQO068W   The value of the *parameter_name* parameter is ignored because the parameter is defined as internal=true. The value is *value_name*.

Explanation: The specified value of the parameter is ignored because it is defined as internal=true.

System action: Processing continues.

User response: Ensure that information was specified correctly on the Discover Customized Product Information panel.

CCQO069W   The Discover EXEC did not find the *parameter_name* parameter in the LPAR metadata. The record was not processed.

Explanation: The specified parameter is missing from the LPAR metadata.

System action: Processing continues.

User response: Ensure that information was specified correctly on the Discover Customized Product Information panel.

CCQO070W   The *record_type* Discover record contains an incorrect delimiter between the Environment section and the Data section. The record was not processed.

Explanation: Tools Customizer found an incorrect delimiter between the Environment section and the Data section.

System action: None.

User response: No action is required.

CCQO071W   The *member_name* member could not be found in the *data_set_name* Discover data set.

Explanation: Tools Customizer could not find the specified Discover data set.

System action: None.

User response: No action is required.

CCQO072S **The** *member_name* **discover metadata member was not found in the** *data_set_name* **metadata data set.**

Explanation: Tools Customizer could not find the specified metadata member in the data set.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQO073E **The** *member_name* **discover metadata member is not valid because the default length for the** *element_name* **parameter element exceeds the length of the parameter. The default length is** *default_length*, **and the specified length is** *specified_length*. **The default length will be truncated accordingly.**

Explanation: The default length for the specified parameter element is longer than the parameter.

System action: Processing continues.

User response: No action is required.

CCQO074S **The content of the** *member_name* **discover metadata member is not valid. The value of the** *attribute_name* **attribute in the** *element_name* **element is not valid. The value of the attribute is** *value_name*.

Explanation: The specified value is not valid.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQO075W **The** *configuration_ID* **configuration ID in the** *record_name* **Discover record is incorrect. The record was not processed.**

Explanation: The specified configuration ID is not correct.

System action: Processing continues.

User response: No action is required.

CCQO076W **The** *configuration_ID* **configuration ID cannot contain more than** *maximum_number* **characters. The record was not processed.**

Explanation: The specified configuration ID contains too many characters.

System action: Processing continues.

User response: No action is required.

CCQO077S **The discover metadata member was not found in the** *data_set_name* **component data set that is part of the** *data_set_name* **pack.**

Explanation: The discover metadata member was not found in the specified component data set.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQO080I *Product_name* **does not support the Discover process.**

Explanation: The specified product does not support the Discover process.

System action: None.

User response: No action is required.

CCQP000E **The value of the** *mode_name* **DB2 mode is not valid for the** *level_name* **DB2 level.**

Explanation: The specified DB2 mode is not valid for the DB2 level.

System action: Processing stops.

User response: Specify a valid DB2 mode for the DB2 level.

CCQP001E **The value of the** *mode_name* **DB2 mode is missing.**

Explanation: The specified DB2 mode is not defined.

System action: Processing stops.

User response: Specify a value for the DB2 mode.

CCQP002E **The value of the** *mode_name* **DB2 level is missing.**

Explanation: The specified DB2 level is not defined.

System action: Processing stops.

User response: Specify a value for the DB2 level.

CCQP003E **The value of the** *level_name* **DB2 level is not valid.**

Explanation: The specified DB2 level does not have a valid name.

System action: Processing stops.

User response: Specify a valid value for the DB2 level.

CCQP004S   **The** *parameter_name* **parameter does not exist in the CCQ$$DB2 DB2 parameter metadata member.**

Explanation: The CCQ$$DB2 DB2 parameter metadata member does not contain the specified parameter.

System action: Processing stops.

User response: See "Gathering diagnostic information" on page 902. Contact IBM Software Support.

CCQP005E   **The value of the** *subsystem_ID* **DB2 SSID is missing.**

Explanation: The specified DB2 SSID is not defined.

System action: Processing stops.

User response: Specify a valid value for the DB2 SSID.

CCQP006E   **The value of the** *group_attach_name* **DB2 group attach name is missing.**

Explanation: The specified DB2 group attach name is not defined.

System action: Processing stops.

User response: Specify a valid DB2 group attach name.

CCQQ000E   **Specify a valid metadata library. Each qualifier of the library must start with an alphabetic character and must be 1-8 alphanumeric characters. The library name must be 1-44 characters.**

Explanation: The metadata library was not specified in the correct format. The high-level qualifier must contain alphanumeric characters, and the first character cannot be numeric. The name cannot contain wildcard characters, such as asterisks (*) and percent signs (%).

System action: Tools Customizer prompts for the correct library name.

User response: Specify a library in the correct format. If the message was issued on the Specify the Metadata Library (CCQPHLQ) panel, specify the product metadata library. The name of this library is *hlq*.SADBDENU.

Do not specify the Tools Customizer metadata library, which is *hlq*.SCCQDENU.

CCQQ001E   **The** *data_set_name* **data set name that was specified for the metadata library was not found.**

Explanation: The data set does not exist, or the data set name was written in the incorrect format. The high-level qualifier must contain alphanumeric characters, and the first character cannot be numeric. The name cannot contain wildcard characters, such as asterisks (*) and percent signs (%).

System action: Tools Customizer prompts for the correct data set name.

User response: Specify a data set name in the correct format.

CCQQ002E   **The data set name that was specified for the** *library_name* **metadata library cannot be opened.**

Explanation: Tools Customizer could not open the data set.

System action: Tools Customizer prompts for an available data set.

User response: Ensure that the specified data set is available for Tools Customizer to open it.

CCQQ003E   **The** *data_set_name* **data set name that was specified for the metadata sample library is not valid. The data set must be in the following format:** `HLQ.S`*xxx*`SAMP.`

Explanation: The specified data set name was not specified in the correct format.

System action: None.

User response: Specify the data set name in the following format: `HLQ.S`*xxx*`SAMP`, where *xxx* is the three-character prefix for the product.

CCQQ004E   **The** *data_set_name* **data set is being used by another user. Try again when the data set is not being used.**

Explanation: Another user is using the specified data set.

System action: None.

User response: Ensure that the specified data set is not being used.

CCQQ009E   **The** *data_set_name* **data set name that was specified for the metadata library is not valid because the data set is empty.**

Explanation: The specified data set is empty.

System action: Tools Customizer prompts for an available data set.

User response: Ensure that the specified data set is available for Tools Customizer to open it.

CCQQ011E **The** *library_name* **metadata library for the component that is part of the** *library_name* **pack was not found in the catalog. The name of the pack is** *pack_name*, **and the name of the component is** *component_name*.

Explanation: The specified metadata library is not in the catalog.

System action: None.

User response: Specify another metadata library.

CCQQ012E **The** *library_name* **metadata library for the component that is part of the** *library_name* **pack cannot be opened.**

Explanation: The specified metadata library cannot be opened.

System action: None.

User response: Ensure that the name of the library is specified correctly.

CCQS000I **Tools Customizer is being invoked for the first time or the previous ISPF session ended before Tools Customizer was exited. In both cases, the fields on this panel are populated with default values. Review these default values or specify new values to be used to customize products or packs.**

Explanation: When you customize a stand-alone product or a solution pack for the first time, or when an ISPF session unexpectedly ends before the ISPF profile is saved, you must specify or review your Tools Customizer user settings.

System action: Processing stops.

User response: Review and accept the default settings, or specify new settings.

CCQS001E **The following command is not valid:** *command_name*.

Explanation: The specified command is not a valid command on the panel.

System action: Processing stops.

User response: Specify a valid command.

CCQS002W **The** *data_set_name* **Discover data set could not be found.**

Explanation: Tools Customizer could not find the specified data set.

System action: The data set will be allocated, and processing continues.

User response: Ensure that the data set name is specified correctly because the data set will be allocated with this name after the values are saved.

CCQS003W **The** *data_set_name* **Discover data set was not found so it was created.**

Explanation: Tools Customizer could not find the specified data set.

System action: Processing continues.

User response: Ensure that the data set name is specified correctly.

CCQS004I **The settings were saved.**

Explanation: The settings that you changed were saved.

System action: Processing continues.

User response: No action is required.

CCQS006W **The length of a qualifier for the** *data_set_name* **customization library data set exceeds 26 characters.**

Explanation: The qualifier for the customization library data set is too long. The qualifier cannot exceed 26 characters.

System action: Processing continues.

User response: Specify a qualifier that is 26 characters or less.

CCQS007E **The discover data set** *data_set_name* **could not be opened with the** *option-type* **option.**

Explanation: The specified option could not open the Discover data set.

System action: None.

User response: Specify a data set to which you have WRITE access.

CCQS008E **An error occurred while the** *data_set_name* **Discover data set was being created.**

Explanation: While the specified data set was being created, an error occurred.

System action: Processing continues.

User response: Ensure that you have WRITE authority access to this data set.

CCQS010E   The customization library qualifier is not valid.

Explanation: The customization library qualifier that was specified is not valid.

System action: None.

User response: Specify a valid qualifier for the customization library.

CCQS011E   The group attach option is not valid.

Explanation: The group attach option that was specified is not valid.

System action: None.

User response: Specify a valid option for the group attach option.

CCQS012E   The Tools Customizer metadata library is not valid.

Explanation: The metadata library that was specified is not a valid data set.

System action: None.

User response: Specify a valid data set for the metadata library.

CCQS013E   The Discover data set is not valid.

Explanation: The Discover data set that was specified is not a valid data set.

System action: None.

User response: Specify a valid Discover data set.

CCQS014E   The data store data set is not valid.

Explanation: The data set that was specified is not a valid data set.

System action: None.

User response: Specify a valid data store data set.

CCQS015E   Tools Customizer is already running.

Explanation: A session of Tools Customizer is already running in your environment. Only one Tools Customizer session is allowed.

System action: None.

User response: The trace data set is being used. Free the trace data set, and start Tools Customizer again.

CCQS018E   Information on the first line of the job card exceeds 57 characters.

Explanation: The first line of the job card can contain only 57 characters. This character limit includes a continuation character.

System action: Tools Customizer clears the first line of the job card.

User response: Specify information that does not exceed 57 characters on the first line of the job card.

CCQS019E   The required trace data set, *data_set_name*, is currently not accessible.

Explanation: The trace data set must be accessible.

System action: Processing stops.

User response: Ensure that the trace data set is accessible.

CCQS020E   An error occurred while the customization library data set was being created. ALTER authority on the high-level qualifier for the customization library data set is required.

Explanation: To create the customization library data set, ALTER authority on the specified high-level qualifier must be granted.

System action: None.

User response: Ensure that ALTER authority for the specified customization library data set is granted.

CCQS021E   The value *value_name* in the field that contains the cursor position is not valid.

Explanation: The specified value is not valid.

System action: None.

User response: Specify a valid value.

CCQS022E   An error occurred while the customization library data set was being opened. UPDATE authority on the high-level qualifier for the customization library data set is required.

Explanation: To open the customization library data set, UPDATE authority on the specified high-level qualifier must be granted.

System action: None.

User response: Ensure that UPDATE authority for the specified customization library data set is granted.

CCQS023E **An error occurred while the customization library data set was being opened. UPDATE authority on the high-level qualifier for the customization library data set is required.**

Explanation: To open the customization library data set, UPDATE authority on the specified high-level qualifier must be granted.

System action: None.

User response: Ensure that UPDATE authority for the specified customization library data set is granted, or specify a different high-level qualifier for the customization library data set on the Tools Customizer Settings panel.

CCQS024E **An error occurred while the customization library data set was being created. ALTER authority on the high-level qualifier for the customization library data set is required.**

Explanation: To create the customization library data set, ALTER authority on the specified high-level qualifier must be granted.

System action: None.

User response: Ensure that ALTER authority for the specified customization library data set is granted, or specify a different high-level qualifier for the customization library data set on the Tools Customizer Settings panel.

CCQS030E **The following command is not a valid CREATE statement:** *command_statement***.**

Explanation: The specified CREATE command statement is invalid because it contains blanks or alphabetic characters.

System action: Processing stops.

User response: Specify a valid CREATE command statement. The correct syntax is CREATE *nn*, where *nn* is 1 - 99.

CCQS031E **The following command is not a valid CREATE statement:** *command_statement***. The number that can be specified with the CREATE command is 1 - 99.**

Explanation: The specified CREATE command statement is invalid because it contains either 0 or a number greater than 99.

System action: Processing stops.

User response: Specify a valid CREATE command

statement. The correct syntax is CREATE *nn*, where *nn* is 1 - 99.

CCQT000I **The product configuration ID** *copied_configuration_ID* **was successfully copied from** *configuration_ID***.**

Explanation: The specified configuration ID was copied.

System action: None.

User response: No action is required.

CCQT001E **The** *command_name* **line command was specified more than once, which is not allowed.**

Explanation: The specified line command cannot be specified more than one time.

System action: Processing stops.

User response: Specify the line command only once.

CCQT002E **The** *configuration_ID* **configuration ID already exists. Specify a different configuration ID.**

Explanation: The specified configuration ID exists.

System action: Processing stops.

User response: Ensure that the specified configuration ID is unique.

CCQT003I **The product configuration ID** *configuration_ID* **was created.**

Explanation: The specified configuration ID was created.

System action: None.

User response: No action is required.

CCQT004I **The product configuration ID** *configuration_ID* **was removed.**

Explanation: The specified configuration ID was removed.

System action: None.

User response: No action is required.

CCQT005E **The product configuration ID** *configuration_ID* **is not valid. The product configuration ID cannot contain a colon (:).**

Explanation: The specified configuration ID contains a colon (:), but a colon is not valid.

System action: Processing stops.

CCQT006E • CCQT017I

User response: Specify a configuration ID that does not contain a colon.

CCQT006E   The *configuration_ID* **configuration ID exists. Specify a different configuration ID.**

Explanation: The specified configuration ID exists.

System action: Processing stops.

User response: Specify another configuration ID.

CCQT007E   The *configuration_ID* **configuration ID exists but was removed from the list of configurations. To use this configuration ID, you must restore it.**

Explanation: The specified configuration ID exists but was removed from the list of available configuration.

System action: Processing stops.

User response: Specify another configuration ID. To restore the specified configuration ID, issue the CREATE command, and specify the same configuration ID again.

CCQT008E   The *configuration_ID* **configuration ID exceeds** *maximum_number* **characters.**

Explanation: The specified configuration ID contains too many characters.

System action: Processing stops.

User response: Specify another configuration ID that does not exceed the maximum number of characters that was set by DB2 Admin.

CCQT010I   Create request for *configuration_ID* **configuration was cancelled by user.**

Explanation: The request to create the specified configuration was canceled.

System action: Processing stops.

User response: No action is required.

CCQT011I   The *configuration_ID* **configuration was not copied.**

Explanation: The specified configuration was not copied.

System action: Processing stops.

User response: No action is required.

CCQT012I   The *configuration_ID* **configuration was not removed.**

Explanation: The specified configuration was not removed.

System action: Processing stops.

User response: No action is required.

CCQT013I   None of the configurations were copied or removed. All of the previously selected configurations are deselected.

Explanation: The selected configurations were not copied or removed, and they are deselected.

System action: Processing stops.

User response: No action is required.

CCQT014E   Specify Y or N and press Enter to continue, or press End to cancel.

Explanation: A function requires input.

System action: Processing stops.

User response: To continue, specify Y or N and press Enter. Otherwise, press End to cancel.

CCQT015E   The *command_name* **command is not allowed during the process of "Select" configuration line command.**

Explanation: The specified command is not allowed while the line command for selecting configurations is processing.

System action: Processing stops.

User response: Remove the specified line command.

CCQT016I   The *configuration_ID* **configuration was not created**

Explanation: The specified configuration was not created.

System action: Processing stops.

User response: No action is required.

CCQT017I   The *configuration_ID* **configuration was not copied.**

Explanation: The specified configuration was not copied.

System action: Processing stops.

User response: No action is required.

Chapter 26. Troubleshooting   **959**

CCQT018E   Specify Y or N, and press Enter.

Explanation: A function requires input.

System action: Processing stops.

User response: To continue, specify Y or N, and press Enter.

CCQT019I   The select *configuration_ID* **configuration process ended.**

Explanation: The select process for the specified configuration is finished.

System action: Processing stops.

User response: No action is required.

CCQT020E   The *configuration_ID* **configuration was not created because the data store was not accessible.**

Explanation: The specified configuration was not created because the data store could not be accessed.

System action: Processing stops.

User response: Ensure that the data store is accessible and create the configuration again.

CCQT021E   The *configuration_ID* **configuration was not copied because the data store was not accessible.**

Explanation: The specified configuration was not copied because the data store could not be accessed.

System action: Processing stops.

User response: Ensure that the data store is accessible and copy the configuration again.

CCQT025I   The *configuration_ID* **configuration was not updated.**

Explanation: The specified configuration was not updated because the edit process was canceled.

System action: Processing stops.

User response: No action is required.

CCQT027I   The product configuration was successfully updated.

Explanation: The configuration was updated.

System action: Processing continue.

User response: No action is required.

CCQX001S   *Product_name* has already been customized by using values from *data_set_name* data store data set. Switch to the specified data store data set to continue customizing this product.

Explanation: The specified product was customized by using values from the specified data store data set.

System action: Processing stops.

User response: Use the specified data store data set to continue customizing the product.

CCQX002S   *component_name* has already been customized by using values from *data_set_name* data store data set. Switch to the specified data store data set to continue customizing this component.

Explanation: The specified component was customized by using values from the specified data store data set.

System action: Processing stops.

User response: Use the specified data store data set to continue customizing the component.

CCQX011I   *Product_name* was not found.

Explanation: The specified product was not found.

System action: Processing stops.

User response: Specify another product.

Frequently asked questions

Find answers to common questions and solutions to common problems.

Customizing DB2 Admin Tool

1. When I customize the DB2 Admin Tool with the Tools Customizer panels, how can I display help information for the input fields?

   Place the cursor in the input field and press PF1.

2. What value should I specify in the **Customized Table Library** field, which is on the Product Parameters panel.

   If you use the Discover EXEC, specify the same dataset as the one in the **Target Table Library** field.

3. When an input field has the ">" sign and I have a long dataset name, how do I enter the name?

   You can use the EXPAND function to bring up a new panel with a greater field length.

4. Why can't I enter input into a parameter field?

   The field is not editable or available.

   Ensure that the necessary tasks and steps are enabled first.

5. On the Product Parameters panel, when I enable Tasks and Steps, how can I keep the panel from scrolling back to the beginning?

   Place the cursor on the Task/Step you just enabled, and then press Enter. The panel scrolls to the current position.

6. When regenerating customization jobs, do I need to resubmit all jobs?

   When generating customization jobs for first the first time, submit the jobs. However, when you regenerate jobs, you only need to submit the jobs that contain a change.

7. Before calling other products such as Table Editor, and Cloning Tool from DB2 Admin, do I need to customize these other products first?

   Yes, if the products are customizable by TCz.

Chapter 27. Tools Customizer reference

Tools Customizer terminology and data sets

Before you use Tools Customizer, you should understand the Tools Customizer terminology and the data sets that Tools Customizer uses during customization.

Tools Customizer terminology

Tools Customizer uses several unique terms that you should be familiar with before you begin to use Tools Customizer.

Products and components

How an IBM Tool is packaged determines whether it is referred to as a product or as a component in the Tools Customizer documentation and interface. An IBM Tool that is ordered as a stand-alone entity (that is, not as part of a solution pack) is referred to as a product. An IBM Tool that is part of a solution pack is referred to as a component. Some IBM Tools are available in both formats; therefore, the same IBM Tool can be referred to as a product or as a component depending on how it is packaged.

DB2 entry

You can customize DB2 Admin on one or more DB2 entries. A DB2 entry can be any of the following items:

DB2 subsystem

A distinct instance of a relational database management system (RDBMS) that is not part of a data sharing group. An example of a DB2 subsystem name is DB01.

DB2 group attach name

The name that is used by the TSO/batch attachment, the call attachment facility (CAF), DL/I batch, utilities, and the Resource Recovery Services attachment facility (RRSAF) as a generic attachment name. An example of a group attach name is DSG1.

DB2 data sharing member

A DB2 subsystem that is assigned by the cross-system coupling facility (XCF) to a data sharing group. An example of a DB2 data sharing member name is DB02.

Tools Customizer maintains the following lists of DB2 entries:

Associated list

The list of DB2 entries that are associated with DB2 Admin. If the product to be customized requires DB2 entries, you can customize DB2 Admin only on DB2 entries that are in the associated list. When you customize DB2 Admin, this list is displayed in the DB2 Entries, Associations, and Parameter Status section of the Customizer Workplace panel.

You can add and copy DB2 entries to the associated list. When you add or copy DB2 entries to the associated list, the entries are associated with DB2 Admin.

Master list

The list of all DB2 entries that are defined but are not associated with DB2 Admin. Tools Customizer obtains information about these DB2 entries either from entries that were created manually or from the customizations of other products that were discovered. If you remove a DB2 entry from the associated list, the DB2 entry is added to the master list. When you create a new DB2 entry, it is added to the master list, and when you associate the new entry with DB2 Admin, it is removed from the master list and added to the associated list. The master list is displayed on the Associate a DB2 Entry for Product panel.

If the associated list does not have the DB2 entries on which you want to customize DB2 Admin, you can associate existing entries from the master list to the associated list.

You can create new DB2 entries and copy existing entries to the master list.

High-level qualifier
The high-level qualifier is considered to be all of the qualifiers except the lowest level qualifier. A high-level qualifier includes a mid-level qualifier.

Product parameters
Parameters that are specific to DB2 Admin. These parameters are defined by DB2 Admin and are stored in a data member that is defined by DB2 Admin.

LPAR parameters
Parameters on the local LPAR that are required to customize DB2 Admin. These parameters are defined by Tools Customizer and are stored in an LPAR parameter data member.

DB2 parameters
Parameters for a DB2 entry. These parameters are defined by Tools Customizer and are stored in a DB2 parameter data member.

Status type

Product, LPAR, and DB2 entry status type

After you specify the product that you want to customize, the product, the LPAR, and the DB2 entries have a status. The status is partly based on whether required parameters are defined. For some products, LPAR parameters or DB2 parameters might not be required. In these cases, the status is Not Required.

To customize DB2 Admin, all of the required parameters must be defined.

If required parameters for the the product parameters, LPAR parameters, or DB2 parameters are not defined, the status of the parameters is Incomplete. Define values for parameters by manually editing them or by generating the customization jobs and specifying values for all of the required parameters that are displayed on the panels.

When values for all of the required parameters are defined, the status is Ready to Customize. Customization jobs can be generated only when all of the required parameters are defined and the status is Ready to Customize or Customized for the product parameters, LPAR parameters, and DB2 parameters for the DB2 entries on which DB2 Admin will be customized.

The following table shows the meaning of the status types. Each status is defined differently for each type of parameter.

Table 28. Status types for the product, the LPAR, and the DB2 entries

Status	Product	LPAR	DB2 entries
Incomplete	The required product parameters are not defined, or the required product parameters are defined but LPAR parameters, DB2 parameters, or both are not defined.	The required parameters are not defined.	The required parameters are not defined.
Discovered	The product parameter definitions were discovered by using the product Discover EXEC.	N/A	N/A
Ready to Customize	The required product, LPAR, and DB2 parameters are defined, the status is Ready to Customize or Customized for the LPAR and at least one associated DB2 entry. You can generate the customization jobs.	The required LPAR parameters are defined or LPAR parameters are not required.	The required DB2 parameters are defined or DB2 parameters are not required.
Customized	The jobs are customized on the local LPAR.	The jobs are customized for the product or for all of the associated DB2 entries on the local LPAR.	The jobs are customized for the DB2 entry.
Errors in Customization	N/A	N/A	Errors occurred while the customization jobs were being generated.
Not Required	N/A	LPAR parameters are not required.	DB2 parameters are not required.

Related tasks:

"Creating and associating DB2 entries" on page 84
You can create new DB2 entries and associate them with DB2 Admin.

"Copying DB2 entries" on page 93
You can copy associated and not associated DB2 entries to other DB2 entries or to new DB2 entries.

"Removing DB2 entries" on page 95
You can remove DB2 entries from the associated list.

Data sets that Tools Customizer uses during customization

Tools Customizer uses several unique data sets during the customization process. Familiarize yourself with these data sets before you begin to use Tools Customizer.

Several different data sets are required to customize DB2 Admin with Tools Customizer. These data sets are supplied by DB2 Admin, supplied by Tools Customizer, or allocated by Tools Customizer.

DB2 Admin provides the following data sets:

Metadata library
Contains the metadata for the product to be customized. Tools Customizer uses the metadata to determine which tasks, steps, and parameters to display on the Product Parameters panel, the LPAR Parameters panel, and the DB2 Parameters panel. This data set also contains the templates that Tools Customizer uses to generate the customization jobs.

The metadata library naming convention is *high_level_qualifier*.SADBDENU, where *high_level_qualifier* is all of the segments of the data set name except the lowest-level qualifier.

You specify the metadata library on the Specify the Metadata Library panel. READ access to this data set is required.

Discover EXEC library
Contains the DB2 Admin Discover EXEC. When you customize DB2 Admin, you can use the Discover EXEC to automatically retrieve and store product information, such as parameter values from an already customized product. Tools Customizer saves the discovered information in the data store.

The default name of the data set is the high-level qualifier for the metadata library plus a lowest-level qualifier. For DB2 Admin, the lowest-level qualifier is SADBEXEC. You can change the default value on the Discover Customized Product Information panel. EXECUTE access to this data set is required.

Tools Customizer provides the following data sets:

Tools Customizer metadata library
Contains the metadata for the DB2 and LPAR parameters that are required to customize DB2 Admin. Tools Customizer uses the metadata to determine which parameters to display on the DB2 Parameters panel and the LPAR Parameters panel. In addition, Tools Customizer uses information in the metadata library to determine whether additional DB2 and LPAR parameters need to be displayed on these panels. As you customize different products, different DB2 and LPAR parameters might need to be defined.

The default name of the data set is DB2TOOL.CCQ110.SCCQDENU. You can change the default value on the Tools Customizer Settings panel. READ access to this data set is required.

Tools Customizer table library
Stores information about jobs that are customized. Job information that is stored includes a description of the job, its member name and template name, the SSID, group attach name, and when the job was generated.

The default name of the data set is DB2TOOL.CCQ110.SCCQTENU. WRITE access to this data set is required.

Tools Customizer requires that the following data sets exist during the customization process. If the data sets do not exist, Tools Customizer automatically allocates them.

Discover output data set

Contains the output that is generated when you run the DB2 Admin Discover EXEC. The DB2 Admin Discover EXEC retrieves the metadata and values for the parameters from a previous customization of DB2 Admin.

The default name of the data set is DB2TOOL.CCQ110.DISCOVER. You can change the default value on the Tools Customizer Settings panel or the Discover Customized Product Information panel. WRITE access to this data set is required.

Data store data set

Contains product, LPAR, and DB2 parameter values, and DB2 entry associations. Tools Customizer uses this data set to permanently store all information that is acquired about the product, DB2 subsystems or data sharing groups, and LPAR when you customize products on the local LPAR.

The default name of the data set is DB2TOOL.CCQ110.DATASTOR. You can change the default value on the Tools Customizer Settings panel. WRITE access to this data set is required.

Customization library

Contains the customization jobs that Tools Customizer generates for DB2 Admin.

Tools Customizer checks whether a customization library name was specified for more than one instance of the same version of the same product. If the same customization library name is specified for more than one product of the same version, the CCQD123E message is issued to prevent you from overwriting previously generated customization jobs. Ensure that you specify unique qualifier for the customization library for each instance of the product.

To customize DB2 Admin, submit the members of the data set in the order in which they are displayed on the Finish Product Customization panel.

The data set naming convention is *hlq*.$*LPAR_name*$.*xyzvrm*, where:

- *hlq* is the value of the **Customization library qualifier** field on the Tools Customizer Settings panel (CCQPSET)
- *LPAR_name* is the four-character LPAR name
- *xyzvrm* is the three-letter product identifier with the version, release, and modification level

For example, the data set name might be DB2TOOL.PRODUCT.CUST.$MVS1$.*XYZ*410.

WRITE access to this data set is required.

Tools Customizer allocates the data sets for the discover output, the data store, and the customization library with the attributes that are shown in the following table:

Table 29. Data set attributes for allocating the Discover output, data store, and customization library data sets

Data set	Organization	Record format	Record length	Block size	Data set name type
Discover output data set	PO	Variable block	16383	32760	LIBRARY

Table 29. Data set attributes for allocating the Discover output, data store, and customization library data sets (continued)

Data set	Organization	Record format	Record length	Block size	Data set name type
Data store data set	PO	Variable block	16383	32760	LIBRARY
Product customization library	PO	Fixed block	80	32720	LIBRARY

Restrictions:

- Multiple users cannot simultaneously share the discover output data set, data store data set, Tools Customizer metadata library, and metadata library.
- You cannot share the data store data set across multiple LPARs with shared DASD or copy the data store data set to another LPAR. Tools Customizer creates many cross-references between product and DB2 associations. Therefore, if you share or copy the data store data set, member names that are empty or that do not exist might be generated.

Chapter 28. System catalog panels

The main system catalog panels are described in this reference information.

Topics:

The System Catalog panel

The System Catalog panel displays objects in the DB2 catalog, database structures, and options for authorizations for objects in the catalog.

Select option 1 on the DB2 Administration Menu to display the System Catalog panel (see Figure 540 on page 970).

Enter one of the object codes on the command line (for example, D for databases). You can limit the information that is returned by entering one or more selection criteria at the bottom of the panel. For example, specifying D402 in the **Name** field limits the search to databases whose names begin with D402. In response to your choices, DB2 Admin creates and executes an SQL statement that searches the DB2 catalog for the object or authorization you have requested.

You can filter your selection by using the **In D/L/H** (database, collection, or schema) field. For example, if you want to display table spaces within a specific database, you select option S and enter the name of a database in the **In D/L/H** field. Or, if you want to display a specific collection in a package, you select option K and specify the collection ID in the **In D/L/H** field.

Recommendation: For optimum performance, specify selection criteria for the following:
- For option T, enter a value for **Owner** or **In D/L/H**.
- Option M can be time-consuming, depending on how many plans and DBRMs you maintain.

When you specify selection criteria, you can change from a LIKE search (a "fuzzy" search) to an exact search, by using an equal sign (=). You can use the LIKE ON and LIKE OFF primary commands to toggle between a "fuzzy" search (LIKE ON) and an exact search (LIKE OFF).

You can save (or not save) your search criteria between DB2 Admin sessions using the SAVE ON and SAVE OFF primary commands. When the SAVE ON command is active, the text "criteria saved" appears on the System Catalog panel. With SAVE ON, the search criteria is restored when you re-enter a DB2 Admin session.

The following figure shows the object options on the System Catalog panel.

```
ADB21 min -------------------- DSNB System Catalog -------------------- 16:17
 Option ===>


 Object options:                                      DB2 System: DSNB
  AO - Authorization options                          DB2 SQL ID: PEDRO
   G - Storage groups               P - Plans
   D - Databases                    L - Collections
   S - Table spaces                 K - Packages
   T - Tables, views, and aliases
   V - Views                        H - Schemas
   A - Aliases for tables and views E - User defined data types
   Y - Synonyms                     F - Functions
   X - Indexes                      O - Stored procedures
   C - Columns                      J - Triggers
   N - Constraints                  Q - Sequences and aliases
  DS - Database structures        DSP - DS with plans and packages
 PDC - DB2 Pending definition changes  GV - Global variables


 Enter standard selection criteria (Using a LIKE operator, criteria saved):
 Name  . . . .              > Grantor . . .        >
 Owner . . . .      >        Grantee . . .        >
 In D/L/H  . .              > Switch Catalog Copy . . . N  (N/S/C)
 And/or other selection criteria (option xC shows you columns for option x)
 Column  . . .              > Operator . .      Value . .
```

Figure 540. System Catalog panel (ADB21) – object options

To view the authorization options, choose the AO option. The authorization options are shown in Figure 541 on page 971.

For optimum performance, specify selection criteria for all authorization options (xA) and enter a value for **Grantor** or **Grantee**.

The following figure shows authorization options for the System Catalog panel.

```
ADB21   -------------------- DB2X System Catalog -------------------- 17:35
Option ===>

                                                          More:     +
Authorization options:                         DB2 System: DB2X
  OO - Object options                          DB2 SQL ID: ISTJE
  GA - Storage group auths       PA - Plan authorizations
  DA - Database authorizations   LA - Collection authorizations
  SA - Table space authorizations KA - Package authorizations
  TA - Table authorizations      HA - Schema authorizations
  VA - View authorizations       EA - User defined data type authorizations
  CA - Column authorizations     FA - Function authorizations
  ZA - System authorizations     OA - Stored procedure authorizations
  UA - User authorizations       QA - Sequence authorizations
  RA - Resource authorizations   TR - Trusted contexts
  RO - Roles                     PM - Permissions
  CO - Column masks              GVA - Global variable authorizations




Enter standard selection criteria (Using a LIKE operator, criteria saved):
Name  . . . .              > Grantor . . .          >
Owner . . . .         >    Grantee . . .            >
In D/L/H  . .              > Switch Catalog Copy . . . N   (N/S/C)
And/or other selection criteria (option xC shows you columns for option x)
Column  . . .              > Operator  . .      Value  . .
```

Figure 541. System Catalog panel (ADB21) – authorization options

DB2 Admin will report authorizations based solely on the DB2 catalog. However,
the actual authorization is affected by other sources that are external to the DB2
catalog such as the following:

- Installation authorities specified using DSNZPARM.
- Any external security system, such as RACF.
- Any security product from any other software provider.
- Any impact of the security user exits, even those supplied by IBM.

Option A. Aliases

Use the Aliases panel to display information about the aliases in the DB2 catalog.

Select option A on the System Catalog panel to display the Aliases panel, as shown
in the following figure.

On this panel, you can reverse engineer DB2 objects by using the GEN line
command.

```
DB2 Admin ------- DSNB Aliases for Tables and Views ------- Row 23 to 28 of 28
Command ===>                                           Scroll ===> PAGE

 Line commands:
  L - List  BR - Browse  DC - Describe columns  Drop - Drop Alias  I - Details
  T - Tables  SEL - Select prototyping  DDL - Generate DDL  GEN - Generate SQL
  ? - Show all line commands

 Sel   Name              Schema   RefObject          RefObj   Location
                                  Name               Schema
       *                 *        *                  *        *
 ----- ----------------> -------- ----------------> -------> ----------------
       SYSCHECKDEP       CFSDSN8  SYSCHECKDEP        SYSIBM   DNS8
       SYSCHECKS         CFSDSN8  SYSCHECKS          SYSIBM   DNS8
       SYSCHECKS2        CFSDSN8  SYSCHECKS2         SYSIBM   DNS8
       SYSCOLAUTH        CFSDSN8  SYSCOLAUTH         SYSIBM   DNS8
       SYSCOLDIST        CFSDSN8  SYSCOLDIST         SYSIBM   DNS8
       SYSCOLDISTSTATS   CFSDSN8  SYSCOLDISTSTATS    SYSIBM   DNS8
 ****************************** END OF DB2 DATA ******************************
```

Figure 542. The Aliases panel (ADB21A) – displaying aliases

The fields on this panel are:

Sel
 Input field where you enter one of the line commands listed on the panel.

Name
 Name of the alias.

Owner
 Authorization ID of the owner of the alias.

RefObject Name
 Name of the table or view to which the alias refers.

RefObj Schema
 The schema of the table or view to which the alias refers.

Location
 Location name of the object of the alias. The field is blank for an alias that was
 not defined with a three-part object name.

Option C. Columns

The Columns panel displays the columns in the DB2 catalog.

Select option C on the System Catalog panel (see "The System Catalog panel" on
page 969) to display the Columns panel (see Figure 543 on page 973).

The following figure shows the Columns panel.

```
 ADB21C in -------------------- DDB2X Columns --------------------- Row 1 of 1
Command ===>                                        Scroll ===> PAGE

 Line commands:
  T - Tables  ST - Specific table  X - Indexes  SX - Specific indexes  A - Auth
  GR - Grant  H - Homonyms  I - Interpret  UR - Update runstats  COM - Comment
  LAB - Label  DI - Dist. stats  PST - Part. stats  CM - Mask  CCM - Create mask
  ? - Show all line commands
 Sel Schema   Name               Column Name       Col No Col Type Length N D F
     *        *                  *                        * *             * * * *
 --- -------- ------------------ ------------------ ------ -------- ------ - - -
     DSN8     DSN8ES1_RS_TBL     RS_SEQUENCE            1 INTEGER      4 N N N
     DSN8     DSN8ES1_RS_TBL     RS_EMPNO               2 CHAR        6 N N N
     DSN8     DSN8ES1_RS_TBL     RS_FIRSTNME            3 CHAR       12 N N N
     DSN8     DSN8ES1_RS_TBL     RS_LASTNAME            4 CHAR       15 N N N
     DSN8     DSN8ES1_RS_TBL     RS_SALARY              5 DECIMAL     9 N N N
     DSN8     DSN8ES1_RS_TBL     RS_BONUS               6 DECIMAL     9 N N N
     DSN881SA STAFF              EMPNUM                 1 CHAR        3 N N N
     DSN881SA STAFF              EMPNAME                2 CHAR       20 Y Y N
     DSN881SA STAFF              GRADE                  3 DECIMAL     4 Y Y N
     DSN881SA STAFF              CITY                   4 CHAR       15 Y Y N
     DSN881SA STAFFV1            EMPNUM                 1 CHAR        3 N N N
     DSN881SA STAFFV1            EMPNAME                2 CHAR       20 Y Y N
     DSN881SA STAFFV1            GRADE                  3 DECIMAL     4 Y Y N
     DSN881SA STAFFV1            CITY                   4 CHAR       15 Y Y N
     DSN881SA TESTSTUFF          TESTNO                 1 CHAR        4 Y Y N
     DSN881SA TESTSTUFF          RESULT                 2 CHAR        4 Y Y N
     DSN881SA TESTSTUFF          TESTTYPE               3 CHAR        3 Y Y N
     DSN8810  ACT                ACTNO                  1 SMALLINT    2 N N N
     DSN8810  ACT                ACTKWD                 2 CHAR        6 N N N
     DSN8810  ACT                ACTDESC                3 VARCHAR    20 N N N
     DSN8810  DEMO_UNICODE       LOWER_A_TO_Z           1 CHAR       26 Y Y N
     DSN8810  DEMO_UNICODE       UPPER_A_TO_Z           2 CHAR       26 Y Y N
     DSN8810  DEMO_UNICODE       ZERO_TO_NINE           3 CHAR       10 Y Y N
     DSN8810  DEMO_UNICODE       X00_TO_XFF             4 VARCHAR   256 Y Y N
     DSN8810  DEPT               DEPTNO                 1 CHAR        3 N N N
     DSN8810  DEPT               DEPTNAME               2 VARCHAR    36 N N N
     DSN8810  DEPT               MGRNO                  3 CHAR        6 Y Y N
     DSN8810  DEPT               ADMRDEPT               4 CHAR        3 N N N
     DSN8810  DEPT               LOCATION               5 CHAR       16 Y Y N
     DSN8810  EACT               ACTNO                  1 SMALLINT    2 N N N
```

Figure 543. Columns panel (ADB21C)

The fields on this panel are:

SEL
  Input field where you enter one of the line commands listed on the panel.

SCHEMA
  Schema of the table or view that contains the column

NAME
  Name of the table or view that contains the column.

COLUMN NAME
  Name of the column.

COL NO
  Numerical position of the column in the table or view.

COL TYPE
  Type of column, which is one of the following data types:
  INTEGER
      Large integer
  SMALLINT
      Small integer
  FLOAT
      Floating-point

CHAR
　　Fixed-length character string

VARCHAR
　　Varying-length character string

LONGVAR
　　Varying-length character string

DECIMAL
　　Decimal

GRAPHIC
　　Fixed-length graphic string

VARG　Varying-length graphic string

LONGVARG
　　Varying-length graphic string

DATE　Date

TIME　Time

TIMESTMP
　　Time stamp

BLOB　Binary large object

CLOB　Character large object

DBCLOB
　　Double-byte character large object

ROWID
　　Row ID data type

DISTINCT
　　distinct type

LENGTH
　　Length attribute of the column or, in the case of a decimal column, its
　　precision. The number does not include internal prefixes to record actual
　　length and null state (where these are applicable).

N　This field indicates whether the column can contain null values. This field
　　contains one of the following values:

Y	Yes
N	No

D　Default value for the column. This field contains one of the following values:

N	None
Y	Yes
B	Yes
1–6	User-defined defaults
S	SQLID
U	USER
A	Generated always
D	Generated by default
I	As identity and generated always
J	As identity and generated as default

F　This field indicates whether the column has a field procedure. This field
　　contains one of the following values:

Y	Yes
N	No

Option D. Databases

The Databases panel displays the databases in the DB2 catalog.

Select option D on the System Catalog panel to display the Databases panel, as shown in the following figure.

The following figure shows the Databases panel.

```
 DB2 Admin ------------------ DB2X Databases -------------------- Row 1 of 25
   Command ===>                                               Scroll ===> PAGE

   Commands: GRANT  MIG  DIS  STA  STO  UTIL
   Line commands:
    T - Tables  S - Table spaces  X - Indexes  G - Storage group  ICS - IC status
    DIS - Display database  STA - Start database  STO - Stop database  A - Auth
    ? - Show all line commands
                            Storage  Buffer         Created        Index
   Select Name     Owner    Group    Pool     DBID By        T E  BPool    I
          *        *        *        *           * *         * * *         *
   ------ -------- -------- -------- -------- ------ -------- - - -------- -
          ADBDCH   ADB      ADBGCH   BP1        271 ISTFL2     E BP2       N
          DBEDB1   DPGROTH  SYSDEFLT BP1        272 DPGROTH    E BP2       N
          DBEDB2   DPGROTH  SYSDEFLT BP1        273 DPGROTH    E BP2       N
          DSNDB04  SYSIBM   SYSDEFLT BP1          4 SYSIBM       BP2       N
          DSNDB06  SYSIBM                         6 SYSIBM     E BP0       N
          DSNDB07  DSCGDB2  SYSDEFLT BP1          7 ISTJE     W   BP2      N
          DSNRGFDB DSCGDB2  SYSDEFLT BP1        257 ISTJE      E BP2       N
          DSNRLST  DSCGDB2  SYSDEFLT BP1        256 ISTJE      E BP2       N
          DSN8D81A DSCGDB2  DSN8G810 BP0        258 ISTJE      E BP2       N
          DSN8D81E DSCGDB2  DSN8G810 BP1        260 ISTJE      U BP2       N
          DSN8D81P DSCGDB2  DSN8G810 BP0        259 ISTJE      E BP2       N
          DSN8D81U DSCGDB2  DSN8G81U BP1        261 ISTJE      E BP2       N
          DSQDBCTL DPGROTH  SYSDEFLT BP1        266 DPGROTH    E BP2       N
          DSQDBDEF DPGROTH  SYSDEFLT BP1        267 DPGROTH    E BP2       N
          DSQ1STBB DPGROTH  SYSDEFLT BP1        265 DPGROTH    E BP2       N
          ISTJED   ISTJE    ISTJEG   BP1        269 ISTJE      E BP2       N
          MAPD1    ISTJE    ISTJEG   BP1        276 ISTJE      E BP2       N
          MAPD2    ISTJE    ISTJEG   BP1        277 ISTJE      E BP2       N
          RAADB    DPGROTH  SYSDEFLT BP1        268 DPGROTH    E BP2       N
          RDBIDB1  DPGROTH  SYSDEFLT BP1        262 DPGROTH    E BP2       N
          RDBIDB2  DPGROTH  SYSDEFLT BP1        263 DPGROTH    E BP2       N
          RDBIDB3  DPGROTH  SYSDEFLT BP1        264 DPGROTH    E BP2       N
          TFLDB    ISTFL2   TFLSG    BP1        270 ISTFL2     E BP2       N
          XXXXX    ISTJE    ISTJEG   BP1        274 ISTJE      E BP2       N
          YYYYY    ISTJE    ISTJEG   BP1        275 ISTJE      E BP2       N
   ****************************** END OF DB2 DATA *******************************
```

Figure 544. Databases panel (ADB21D)

The following primary commands are valid on this panel:

GRANT
    Issues a GRANT command on multiple databases.

MIG
    Issues a MIG command on multiple databases.

DIS
    Issues a DB2 DISPLAY command on multiple databases.

STA
    Issues a DB2 START command on multiple databases.

STO
    Issues a DB2 STOP command on multiple databases.

UTIL
    Selects the table spaces for multiple databases for which to generate utility JCL.

If the size of the statements generated by the GRANT, DIS, STA, or STO primary command exceeds 32K (an ISPF limit), you will be prompted to send the statements to a batch job or a work statement list (WSL).

If the number of statements generated by the DIS, STA, or STO primary command exceeds 10, you will be prompted to send the statements to a batch job or a WSL.

Recommendation: Primary commands operate on each row that is displayed in the table. If you want to omit some of the rows before you issue the primary command, use the minus (-) line command to remove rows from the display. The primary commands operate only on rows that are listed.

The fields on this panel are:

SELECT
Input field where you enter one of the line commands listed on the panel.

NAME
Name of the database.

OWNER
Authorization ID of the owner of the database.

STORAGE GROUP
Name of the default storage group for the database. For system databases, this field is blank.

BUFFER POOL
Name of the default buffer pool for the database. For system databases, this field is blank.

DBID
Internal ID for the database.

CREATED BY
Primary authorization ID of the user who created the database.

T Type of database, which is one of the following values:
- **W** Work file
- **T** Temporary database
- **blank** Not a work file database or a temporary database

E Type of encoding, which is one of the following values:
- **E** EBCDIC
- **A** ASCII
- **U** Unicode
- **blank** Work file or temporary database

INDEX BUFFER POOL
Name of the default buffer pool for indexes.

I Implicitly-created database: Y-YES N-NO

Option DS. Database Structures

When you select option DS, the Database Structures panel displays a structured list of objects in the database that you have selected but does not display plans and packages.

Select option DS on the System Catalog panel to display the Database Structures panel, as shown in the following figure. You must enter a value in the Name field prior to selecting the DS option. Otherwise, you will receive the following message: `Invalid for this option`.

The following object types are displayed on the Database Structures panel:
- Databases
- Table spaces
- Tables
- Materialized query tables
- Indexes
- Aliases
- Views on a table
- Synonyms on a table
- Triggers
- Check conditions
- Unique constraints
- Referential constraints (parents)
- Referential constraints (children)

Views on a view and authorizations are not included in this display.

The following figure shows the Database Structures panel without plans and packages displayed.

```
  DB2 Admin --------------- DB2X Database Structures --------- Row 1 to 35 of 35
  Command ===>                                                Scroll ===> PAGE

  Line commands:  S - Show object   DSN - Data sets
                                                    PSID/
  Sel Type   Object Name              Qualifier  DBID ISOBID   OBID Note
      *      *                        *              *      *      * *
  --- ------ ------------------------> -------- ------ ------ ------ -------------
      D----- PJMDBPLN----------------              375      0      0
       S     PJS1                     PJMDBPLN      375      2      1
        T      PJS1T1                 MARINO        375      0      3
         Y        PJS1T1Y1            MARINO          0      0      0
         Y        PJS1T1Y2            MARINO          0      0      0
         CHK      PJCHK1                             375      0     20
        T      PJS1T2                 MARINO        375      0      7
         ALI      PJS1T2A1            MARINO          0      0      0
         X        PJS1T2X1            MARINO        375     14     13
         MQT      PJMMQT1             MARINO          0      0      0
         V        PJS1T2V1            MARINO          0      0      0
         V        PJS1T2V2            MARINO          0      0      0
        T      X.F                    WONG          375      0     17
       S     PJS2                     PJMDBPLN      375      5      4
        T      PJS2T1                 MARINO        375      0      6
         ALI      PJS2T1A1            MARINO          0      0      0
         Y        PJS2T1Y1            MARINO          0      0      0
         Y        PJS2T1YY            MARINO          0      0      0
         UC       PJUCC5                               0      0      0 Unique key
         X        PJS2T1X             MARINO        375     21     19
       S     PJS3                     PJMDBPLN      375      9      8
        T      PJS3T1                 MARINO        375      0     10
         Y        PJS3T1Y1            MARINO          0      0      0
         PAR      PJS3T1FK                             0      0     29
         CHK      PJCHKX                             375      0     18
         X        PJS3T1X             MARINO        375     26     25
         X        PJS3T1X1            MARINO        375     12     11
         V        PJS3T1V1            MARINO          0      0      0
         V        PJS3T1V2            MARINO          0      0      0
       S     PJS4                     PJMDBPLN      375     16     15
       S     PJS5                     PJMDBPLN      375     23     22 Partitioned
        T      PJS5T1                 MARINO        375      0     24
         CHR      PJS3T1FK                             0      0     29
         UC       C1                                   0      0      0 Primary key
         X        PJS5T1X             MARINO        375     28     27
  ***************************** END OF DB2 DATA ******************************
```

Figure 545. Database Structures panel (ADB21DS) without plans and packages displayed

The fields on this panel are:

SELECT

Input field where you enter line command S to show an object.

TYPE

Type of object, which is one of the following:

ALI   Alias

CHK   Check Constraint

CHR   Referential constraint: parent to child

D   Database

J   Trigger

K   Package (shown only for the DSP command)

MQT   Materialized query table (treated as a table when preceded by two blanks in the Type field and as a view when preceded by three blanks)

P   Plan (shown only for the DSP command)

PAR   Referential constraint: child to parent

S   Table Space

T   Table

UC   Unique Constraint

V   View

X   Index

Y   Synonym

OBJECT NAME
Name of the object.

QUALIFIER
DB2 qualifier for the object, if relevant.

DBID
Internal identifier of the database.

PSID/ISOBID
Internal identifier of the table space page set descriptor or index page set descriptor.

OBID
Identifier for the object's internal descriptor.

Option DSP. Database Structures with Plans and Packages

When you select option DSP, the Database Structures panel shows plans and packages that are dependent on the table spaces, tables, views, indexes, aliases, and synonyms.

Select option DSP on the System Catalog panel to display the Database Structures panel, as shown in the following figure, that includes showing the plans and packages that are dependent on the table spaces, tables, views, indexes, aliases, and synonyms.

In the Database Structures panel, plans (P) and packages (K) are indented under the object upon which they are dependent. To eliminate repetitiveness in the display, a dependency on a table is not shown if it is already reported under a view, alias, synonym, or index for the table. Likewise, a dependency for a table space is not shown if it is already reported under a table.

You must enter a value in the Name field prior to selecting the DSP option. Otherwise, you will receive the message, `Invalid for this option`.

The following figure shows the Database Structures panel with plans and packages displayed.

```
DB2 Admin --------------- DB2X Database Structures --------- Row 1 to 35 of 72
Command ===>                                               Scroll ===> PAGE

Line commands:  S - Show object   DSN - Data sets
                                                PSID/
Sel Type   Object Name            Qualifier DBID ISOBID  OBID Note
    *      *                      *             *      *     * *
--- ------ -----------------------> -------- ------ ------ ------ -------------
    D----- PJMDBPLN----------------           375      0      0
     S     PJS1                   PJMDBPLN    375      2      1
      T      PJS1T1               MARINO      375      0      3
       Y       PJS1T1Y1           MARINO        0      0      0
        K         PLISQL          PLISQL        0      0      0
        K         PLISQL          PLISQL3       0      0      0
        K         PLISQL3         PLISQL3       0      0      0
        P         PLISQLP2                      0      0      0
       Y       PJS1T1Y2           MARINO        0      0      0
        K         PLISQL          PLISQL        0      0      0
        K         PLISQL          PLISQL3       0      0      0
        K         PLISQL3         PLISQL3       0      0      0
        P         PLISQLP2                      0      0      0
      CHK    PJCHK1                           375      0     20
       P       PLISQLPL                         0      0      0
      T      PJS1T2               MARINO      375      0      7
      ALI      PJS1T2A1           MARINO        0      0      0
        K         PLISQL          PLISQL        0      0      0
        K         PLISQL          PLISQL3       0      0      0
        K         PLISQL3         PLISQL3       0      0      0
        P         PLISQLP2                      0      0      0
       X       PJS1T2X1           MARINO      375     14     13
      MQT      PJMMQT1            MARINO        0      0      0
        K         PLISQL          PLISQL        0      0      0
        K         PLISQL          PLISQL3       0      0      0
        K         PLISQL3         PLISQL3       0      0      0
        P         PLISQLPM                      0      0      0
       V       PJS1T2V1           MARINO        0      0      0
        K         PLISQL          PLISQL        0      0      0
        K         PLISQL          PLISQL3       0      0      0
        K         PLISQL3         PLISQL3       0      0      0
        P         PLISQLP2                      0      0      0
        P         PLISQLP3                      0      0      0
       V       PJS1T2V2           MARINO        0      0      0
      T      X.F                  WONG        375      0     17
```

Figure 546. Database Structures panel (ADB21DS) with plans and packages displayed

Option E. User-Defined Data Types

Use the Data Types panel to display information about the data types in the DB2
catalog.

Select option E on the System Catalog panel to display the Data Types panel, as
shown in the following figure.

On the Data Types panel, you can reverse engineer DB2 objects.

```
  DB2 Admin ------------------ DB2X Data Types ------------------- Row 1 of 17
Commands: GRANT ARRAY-INFO
Line commands:
 T - Tables  A - Auth  AH - Schema auth  GR - Grant  DROP - Drop  COM - Comment
 I - Interpret  CRE - Create data type  GEN - Generate DDL  DDL - Object DDL
 REP - Report  CP - Copy privileges  RO - Role

                               Source                 M
 S    Schema   Data Type Name  Schema   Source Data Type T     Length  Scale
      *        *               *        *               *          *      *
 ---- -------- --------------- -------- --------------- - ----------- ------
 ____ ULVEMAN  TEST            SYSIBM   INTEGER         T          4      0
 ____ ULVEMAN  ARR             SYSIBM   CHAR            A          8      0
 ____ ULVEMAN  ARRVCI          SYSIBM   CHAR            L          8      0
 ____ ULVEMAN  ARR4            SYSIBM   CHAR            A          8      0
 ____ ULVEMAN  INT_ARR         SYSIBM   INTEGER         A          4      0
 ____ ULVEMAN  ARRINT_INT      SYSIBM   INTEGER         L          4      0
 ____ ULVEMAN  SCALARUDT       SYSIBM   INTEGER         T          4      0
 ____ ULVEMAN  INTARRAY        SYSIBM   INTEGER         A          4      0
 ____ ULVEMAN  ARRTYP1         SYSIBM   INTEGER         A          4      0
 ____ ULVEMAN  MYINTATYPE      SYSIBM   INTEGER         A          4      0
 ____ ULVEMAN  MYCHARATYPE     SYSIBM   VARCHAR         A         20      0
 ***************************** END OF DB2 DATA ******************************
```

Figure 547. Data Types panel (ADB21E)

The layout in Figure 1 is shown the first time the panel is displayed. You can
toggle between displaying data type information (as shown in figure 1) or array
information as shown in the in the following figure:

```
  DB2 Admin ------------------ DB2X Data Types ------------------- Row 1 of 17
  Command ===>                                        Scroll ===> PAGE

  Commands: GRANT TYPE-INFO
  Line commands:
   T - Tables  A - Auth  AH - Schema auth  GR - Grant  DROP - Drop  COM - Comment
   I - Interpret  CRE - Create data type  GEN - Generate DDL  DDL - Object DDL
   REP - Report  CP - Copy privileges  RO - Role

                      M                        Array
  S    Schema   Name  T  Array Cardinality  Ix Type ID  Array Index Length
       *        *     *                  *           *                   *
  ---- -------- -------->  - -------------------- ----------- --------------------
  ____ ULVEMAN  TEST    T                  0           0                   0
  ____ ULVEMAN  ARR     A                  8           0                   0
  ____ ULVEMAN  ARRVCI  L                  0         448                   8
  ____ ULVEMAN  ARR4    A                  4           0                   0
  ____ ULVEMAN  INT_ARR A         2147483647           0                   0
  ____ ULVEMAN  ARRINT_IN L                0         496                   4
  ____ ULVEMAN  SCALARUDT T                0           0                   0
  ____ ULVEMAN  INTARRAY  A               10           0                   0
  ____ ULVEMAN  ARRTYP1   A              100           0                   0
  ____ ULVEMAN  MYINTATYP A               10           0                   0
  ____ ULVEMAN  MYCHARATY A               20           0                   0
  ***************************** END OF DB2 DATA ******************************
```

Figure 548. Array Types panel (ADB21E)

Use the ARRAY-INFO command to show array information , or TYPE-INFO to
show data type information.

The following primary command is valid on this panel:

GRANT

Issues a GRANT command on multiple user-defined data types.

Recommendation: The GRANT command operates on each row that is
displayed in the table. If you want to omit some of the rows before you issue

the command, you can use the minus (-) line command to remove rows from the display. The GRANT command operates only on rows that are listed.

The fields on this panel are:

S   Input field where you enter one of the line commands listed on the panel.

SCHEMA
   Schema of the data type.

DATA TYPE NAME
   Name of the data type.

SOURCE SCHEMA
   Schema of the source data type.

SOURCE DATA TYPE
   Name of the source data type for this distinct data type.

MT   Metatype: specify T for Distinct, A for Array, or L for an Associative array.

LENGTH
   Maximum length for the data type, or precision for distinct types.

SCALE
   Scale for distinct data types, based on the built in decimal type.

Creating an array data type

Use the Create Array Type panel to create a new array type.

Procedure

1. Select option CE on the Create/Drop/Label/Comment On Objects panel. The Create Type panel is displayed, as shown in the following figure.

```
ADB26CE n ----------------- DB2 Create Type --------------------- 13:4
+—————————————————————————————————————————————————————————————————————+
| ADB2CONF  -- DSNB CREATE TYPE - choice --------------------------- 13:41 |
|                                                                       |
| Specify which user-defined                                            |
| data type to create.                                                  |
|                                                                       |
|                                                                       |
|                                                                       |
| Select a choice                                                       |
|   1.  Distinct type                                                   |
|   2.  Array type                                                      |
|                                                                       |
|                                                                       |
|                                                                       |
|                                                                       |
|  F1=Help     F2=Split    F3=Exit     F9=Swap    F12=Cancel            |
+—————————————————————————————————————————————————————————————————————+
```

Figure 549. Create Type panel

2. Select option 2 for ARRAY TYPE. The Create Array Type panel is displayed, as shown in the following figure.

```
ADBP6CAT  ----------------- DSNB Create Array Type    ----------------- 13:42
Command ===>

 CREATE TYPE

 Schema . . . . . . .          >          (Default is VNDEJB)
 Name . . . . . . . .                     > (? to look up)

 AS
   Source type. . . .                     > (Built-in data type)
   Length . . . . . .            (Precision for TIMESTAMP and DECIMAL)
   Scale  . . . . . .            (For DECIMAL only)

   FOR ? DATA . . . .            (BIT, SBCS, or MIXED)

   CCSID  . . . . . .            (optional: ASCII, EBCDIC, or UNICODE)

   WITH TIME ZONE . .            (Yes/No - for TIMESTAMP only)

 ARRAY
   Constant . . . . .            (integer value from 1 to 2147483647)
     or
   Array subtype  . .            (INT, VARCHAR or blank)
    Length . . . . .             (for VARCHAR only)
    CCSID  . . . . .             (optional: ASCII, EBCDIC, or UNICODE)
    FOR ? DATA . . .             (optional: BIT, SBCS, or MIXED)
```

Figure 550. Create Array Type panel (ADBP6CAT)

3. Specify the following values for the array type:
    a. In the **Schema** field, enter the schema.
    b. In the **Name** field, enter the name.
    c. In the fields within the **AS** area, enter the information that goes inside the brackets of an AS clause.
        *   In the **Source type** field, enter the name of the built-in data type.
        *   If specifying a TIMESTAMP or DECIMAL, enter the length in the **Length** field.
        *   If specifying a DECIMAL, enter the scale in the **Scale** field.
        *   In the **FOR / DATA** field, BIT, SBCS, or MIXED.
        *   In the optional **CCSID** field, ASCII, EBCDIC, or UNICOD.
        *   If specifying a TIMESTAMP, enter YES or NO in the **WITH TIME ZONE** field.
    d. In the fields within the **ARRAY** area, enter the following fields. Array subtype and Constant are mutually exclusive. An error message is returned if both array subtype and constant are non-blank. Leave the fields blank if you want to use the Constant default value of 2147483647.
        *   In the **Array** subtype field, enter INT or VARCHAR.
        *   In the **Constant** field, enter an integer value from rom 1 to 2147483647..
        *   If specifying a varchar,, enter the length in the **Length** field.
        *   If specifying a varchar, optionally enter ASCII, EBCDIC, or UNICODE in the **CCSID** field.
        *   If specifying a varchar, optionally enter BIT, SBCS, or MIXED in the **FOR ? DATA** field.

Option F. Functions

Use the Functions panel to display information about the functions in the DB2 catalog.

Select option F on the System Catalog panel to display the Functions panel, as
shown in the following figure.

```
ADB21F in -------------------- DB2X Functions -------------- Row 1 to 9 of 415

Commands: GRANT  VERSION
Line commands:
 AH - Schema auth  A - Auth  DROP - Drop  AL - Alter  K - Package  PA - Parms
 RT - Return type  DIS - Display  STO - Stop  STA - Start  GR - Grant
 ? - Show all line commands
                                                         D     S
                      External   Specific         F      E E C Q S P E
Sel  Schema   Name    Name       Name        I O T  Parms T A F L R T S
     *        *       *          *           * * *        * * * * * * * *
---- -------- -------> -------- ------------------ - - - ------ - - - - - - - -
     DSNADM   ADMIN_TA DSNADMTL  ADMIN_TASK_LIST    E T     0 N E N R N S D
     DSNADM   ADMIN_TA DSNADMTO  ADMIN_TASK_OUTPUT  E T     2 N E N R N S D
     DSNADM   ADMIN_TA DSNADMTS  ADMIN_TASK_STATUS  E T     0 N E N R N S D
     DSNADM   ADMIN_TA DSNADMTH  ADMIN_TASK_STATUSH E T     1 N E N R N S D
     DB2MQ    MQREAD   DSN2RD    DSN2RD             E S     3 N E N R Y S D
     DB2MQ    MQREAD   DSN2RD0   DSN2RD0            E S     0 N E N R Y S D
     DB2MQ    MQRECEIV DSN2XC2R  DSN2XC2R           E T     3 N E N R Y S D
     DB2MQ    MQREADCL DSN2RDC   DSN2RDC            E S     3 N E N R Y S D
     DB2MQ    MQREADCL DSN2RDC0  DSN2RDC0           E S     0 N E N R Y S D
```

Figure 551. Functions panel (ADB21F)

The following primary command is valid on this panel:

GRANT
    Issues a GRANT command on multiple functions.

    Recommendation: The GRANT command operates on each row that is
    displayed in the table. If you want to omit some of the rows before you issue
    the command, you can use the minus (-) line command to remove rows from
    the display. The GRANT command operates only on rows that are listed.

The fields on this panel are:

SEL
    Input field where you enter one of the line commands listed on the panel.

SCHEMA
    Schema of the function.

NAME
    Name of the function.

EXTERNAL NAME
    Load module name for the stored procedure. This field is blank if it is not an
    external or user-defined function.

VERSION/EXTERNAL
    Toogles to a view which includes either the External Name column or the
    Version and Active columns.

A   Identifies the active version of a routine.

SPEC NAME
    The specific name of the function.

I   Indicates if the routine is an inline function. Indicate Yes or No.

O   Origin of the function, which is one of the following values:
    E   External

	U	Sourced
	S	System generated
	Q	SQL

FT Function type, which is one of the following types:
  C Column
  S Scaler
  T Table

PARMS
  Number of parameters for the function.

DET
  This field indicates whether the external function is deterministic (that is, returns the same result when called using the same parameters). This field contains one of the following values:
  Y Yes
  N No
  blank The routine is a function, but not an external function.

EA This field indicates whether the external function changes the state of an object that DB2 does not manage. This field contains one of the following values:
  E Yes
  N No
  blank The routine is a stored procedure.

CF Cast function, which is one of the following values:
  Y Yes
  N No

SQL
  This field indicates whether SQL statements are allowed, which is one of the following values:
  N Contains no SQL statements
  C Contains SQL statements
  R Reads SQL data
  M Modifies SQL data
  blank Not applicable.

SR This field indicates whether the program should remain resident when it ends. This field contains one of the following values:
  Y Program remains resident
  N Program does not remain resident
  blank Not external or user-defined function.

PT Program type, which is one of the following types:
  M Main
  S Subroutine
  blank Not external or user-defined function.

ES External security, which is one of the following values:
  D DB2 address space user
  U User
  C Definer
  blank Not external or user-defined function.

Option G. Storage Groups

The Storage Groups panel displays the storage groups in the DB2 catalog.

Select option G on the System Catalog panel to display the Storage Groups panel, as shown in the following figure.

```
  ADB21G in ----------------- DB2X Storage Groups ------------ Row 1 to 10 of 26

  Line commands:
   D - Databases  S - Table spaces  X - Indexes  VOL - Volumes  I - Interpret
   GR - Grant  DROP - Drop  CRE - Create  AL - Alter  UT - Utility  A - Auth
   DDL - Generate DDL  GEN - Generate SQL  REP - Report  RO - Role
   ? - Show all line commands

  Select Name      Owner     VCAT          Space Statistics time
         *         *         *             *  *
  ------ --------  --------  --------  ----------- -------------------------
         ADBGCH    ADB       DB2X            0 0001-01-01-00.00.00.000000
         DSN8G81U  DSCGDB2   DB2X            0 0001-01-01-00.00.00.000000
         DSN8G810  DSCGDB2   DB2X            0 0001-01-01-00.00.00.000000
         ISTJEG    ISTJE     DB2X            0 0001-01-01-00.00.00.000000
         SYSDEFLT  DSCGDB2   DB2X            0 0001-01-01-00.00.00.000000
         TFLSG     ISTFL2    DB2X            0 0001-01-01-00.00.00.000000
```

Figure 552. Storage Groups panel (ADB21G)

The fields on this panel are:

SELECT
Input field where you enter one of the line commands listed on the panel.

NAME
Name of the storage group.

OWNER
Authorization ID of the owner of the storage group.

VCAT
Name of the VSAM or ICF catalog.

SPACE
Kilobytes (KB) of storage allocated for the storage group as determined by the STOSPACE utility the last time it was run. A value of -1 indicates that the utility has never been run.

Statistics Time
The timestamp of when the Space field was last updated.

Option GV. Global Variables

Use the Global Variables panel to display information about the global variables in the DB2 catalog.

Select option GV on the System Catalog panel to display the Global Variables panel, as shown in the following figure.

```
ADBP1GV n ---------------- DSNB Global Variables ---------- Row 1 to 11 of 325

Line commands:
 I - Interpretation  A - Auth  GEN - Generate DDL  DDL - Object DDL
 CRE - Create  COM - Comment  ALT - Alter  DROP - Drop  DO - Dependent objects
 ? - Show all line commands
                                    Data    Max
Select Schema   Name                Type    Length Scale Default Text
       *        *                   *          *      * *
------ -------> ----------------- -------- <----- ------ --------------------
       SYSIBM   CLIENT_IPADDR       CHAR        39      0 NULL
       SYSIBMAD GET_ARCHIVE         CHAR         1      0 'N'
       SYSIBMAD MOVE_TO_ARCHIVE     CHAR         1      0 'N'
       VNDRG    VAR1                INTEGER      4      0
       VNDRG    VAR2                VARCHAR    100      0
       VNDRG    VWINT               INTEGER      4      0
       VNDRG    TEXT                VARCHAR    128      0
       VNDRG    VARCHAR128          VARCHAR    128      0
       GVAR     TEST                VARCHAR    128      0
       GVAR     TESTFUNC            VARCHAR    128      0
       VNDRG    GINT                INTEGER      4      0
```

Figure 553. Global Variables panel (ADBP1GV)

The following primary commands are valid on this panel:

I   Interpretation. Provides detailed information about a specific global variable.

A   Authorization. display information about the users who grant privileges to global variables, and information about the users who hold the privileges.

GEN
Generate DDL. Generate SQL statements.

DDL
Object DDL

CRE
Create.

COM
Comment. Object DDL

ALT
Alter. Object DDL

DROP
Comment. Object DDL

DO   Dependent objects. Object DDL

The following fields are displayed on this panel:

Schema
The schema of the global variable.

Name
The name of the global variable.

Data Type
The name of the data type.

Max Length
The maximum length of the global variable.

Scale
The scale of the global variable.

Default Text
> The text of the default value of the global variable.
>
> If the text is truncated, type EXPAND on the primary command line, position the cursor on the default text field, and press Enter to display all of the text.

Option H. Schemas

Use the Schemas panel to display the schemas in the DB2 catalog.

Select option H on the System Catalog panel to display the Schemas panel, as shown in the following figure.

On the Schemas panel, you can reverse engineer DB2 objects.

```
   ADB21H in -------------------- DB2X Schemas -------------- Row 1 to 10 of 141

 Line commands:
  E - Data type   F - Function  J - Trigger  O - Stored procedure  A - Auth
  GR - Grant  GEN - Generate DDL  REP - Report  Q - Sequence  GV - Gbl. Variable
  CP - Copy privileges

                Number of  Number of  Number of  Number of  Number of  Number of
 S   Schema    Data Types  Functions Procedures  Triggers  Sequences  Variables
     *              *          *          *          *          *          *
 --- ------->  ---------- ---------- ---------- ---------- ---------- ----------
     A                  0          0          0          0          0          1
     ADB                0          0          2          1          0          2
     ADMF001            0          0          0          0          0          1
     ADMF002            0          0          0          0          0          1
     ADMINO             1          2          0          0          0          0
     ARRAY_TE           1          2          1          0          0          0
     ARRAY_TE           4          8          0          0          0          0
     ASWD               0          0          0          0         38          0
     AWDV               0         11         99        113         32          0
     B                  0          0          0          0          0          1
```

Figure 554. Schemas panel (ADB21H)

The fields on this panel are:

S   Input field where you enter one of the line commands listed on the panel.

Schema
> Schema of the data type.

Number of Data Types
> Number of distinct data types defined in this schema.

Number of Functions
> The number of user-defined functions and implicitly-defined functions in this schema.

Number of Procedures
> Number of stored procedures defined in this schema.

Number of Triggers
> Number of table triggers defined in this schema.

Number of Sequences
> Number of sequences defined in this schema. To view the sequences, issue the Q line command against a schema that contains a number of sequences in the Number of Sequences column. The Sequence Objects panel (ADB21Q) is displayed.

Number of Variables
Number of variables defined in this schema. To view the global variables, issue the gv line command against a schema that contains a number in the Number of Variables column. The Global Variables panel (ADBP1GV) is displayed

Option J. Triggers

Use the Triggers panel to display information about the triggers in the DB2 catalog.

Select option J on the System Catalog pane to display the Triggers panel, as shown in the following figure.

```
ADB21J in ------------------- DB2X Triggers ----------------- Row 1 to 1 of 1

Line commands:
 D - Database  T - Table  K - Package  A - Schema auth  I - Interpretation
 GEN - Generate DDL  Drop - Drop  COM - Comment  CRE - Create  AL - Alter
 ? - Show all line commands
                                  Table/    Table/
                                  View      View                    Created
S    Schema   Name     Owner     Schema    Name              T E G By
     *        *        *         *         *                 * * * *
---- -------- ------->  -------- -------- ------------------ - - - --------
AL   DSNIBMTS CONNECTI DB2ADM    SYSIBMTS SYSTEXTCONNECTINFO B I R DB2ADM
****************************** END OF DB2 DATA ******************************
```

Figure 555. Triggers panel (ADB21J)

The fields on this panel are:

S   Input field where you enter one of the line commands listed on the panel.

SCHEMA
Name of the schema.

NAME
Name of the trigger.

OWNER
Authorization ID of the owner of the trigger.

TABLE/VIEW SCHEMA
Schema of the table or view to which this trigger applies.

TABLE/VIEW NAME
Name of the table or view to which this trigger applies.

T   Trigger time, which is one of the following values:
    A   After
    B   Before
    I   Instead of

E   Trigger event, which is one of the following values:
    I   Insert
    U   Update
    D   Delete

G   Granularity of the trigger, which is one of the following values:
    R   For each row
    S   For each statement

CREATED BY
Primary authorization ID of the user who created the trigger.

Option K. Packages

The Packages panel displays the packages in the DB2 catalog.

Select option K on the System Catalog panel to display the Packages panel, as shown in the following figure.

```
  ADB21K in ------------------- DB2X Packages --------------- Row 1 to 30 of 104
  Command ===> _____ Scroll ===> PAGE

  Commands:  BIND  REBIND  FREE  BINDOPT  VERSIONS  GRANT  ALL  PLANMGMT
  Line commands:
   DP - Depend  A - Auth  T - Tables  V - Views  X - Indexes
   S - Table spaces  Y - Synonyms  Q - Sequences  RB - Rebind  F - Free  B - Bind
   BC - Bind copy  EN -Enab/disab con  PL - Package lists  P - Local plans
   GR - Grant  I - Interpret  SQ - SQL in package  LP - List PLAN_TABLE
   LPA - List all PLAN_TABLE  VE - Versions  D - Databases  RO - Role
   DET - Package details
                                                         V I V O Quali-  R E D
  S  Collection         Name      Owner    Version (trunc) D S A P fier   L X R
     *                  *         *        *              * * * * *       * * *
  -- ------------------ --------  -------- ---------------- - - - - -------- - - -
  __ DSNTIAD            DSNTIAD   LLEGARD                   R S Y Y LLEGARD    N
  __ DSNREXX            DSNREXX   LLEGARD  V10R1            B S Y Y LLEGARD    N
  __ DSNREXUR           DSNREXX   LLEGARD  V10R1            B U Y Y LLEGARD    N
  __ DSNREXCS           DSNREXX   LLEGARD  V10R1            B S Y Y LLEGARD    N
  __ DSNREXRS           DSNREXX   LLEGARD  V10R1            B T Y Y LLEGARD    N
  __ DSNREXRR           DSNREXX   LLEGARD  V10R1            B R Y Y LLEGARD    N
  __ DSNTIAP            DSNTIAP   DB2ADM                    R   Y Y DB2ADM     N
  __ DSNESPCS           DSNESM68  DB2ADM                    R S Y Y DB2ADM     N
  __ DSNESPRR           DSNESM68  DB2ADM                    R R Y Y DB2ADM     N
  __ DSNESPUR           DSNESM68  DB2ADM                    R U Y Y DB2ADM     N
  __ DSNEDCL            DSNECP68  DB2ADM   V10R1            R S Y Y DB2ADM     N
  __ DSNUTIL            DSNUGSQL  DB2ADM   V10R1            B S Y Y DB2ADM     N
  __ DSNUT101           DSNUGSQL  DB2ADM   V10R1            B S Y Y DB2ADM     N
  __ DSNADM             DSNADMJF  LLEGARD  V10R1            R S Y Y LLEGARD    N
  __ DSNADM             DSNADMTA  LLEGARD  V10R1            R S Y Y LLEGARD  C N
  __ DSNADM             DSNADMTR  LLEGARD  V10R1            R S Y Y LLEGARD  C N
  __ DSNADM             DSNADMTU  LLEGARD  V10R1            R S N Y LLEGARD  C N
```

Figure 556. Packages panel (ADB21K)

The following primary commands are valid on this panel:

BIND

Issues a BIND command on multiple packages. When you attempt to bind more than 20 packages, DB2 Admin prompts you to specify either a work statement list or a batch job to complete the processing.

BINDOPT

Displays the Bind Options panel. From the panel, you can choose bind and rebind options that are not in the DB2 catalog records.

FREE

Issues a FREE command on multiple packages. When you attempt to free more than 20 packages, DB2 Admin prompts you to specify either a work statement list or a batch job to complete the processing.

REBIND

Issues a REBIND command on multiple packages. When you attempt to rebind more than 20 packages, DB2 Admin prompts you to specify either a work statement list or a batch job to complete the processing.

When you specify REBIND, the resulting BIND command contains only the package name. Specify REBIND FULL. If you want the resulting BIND command to contain the package name and all of the parameters.

Note: You can issue the BIND, REBIND, or FREE command on multiple packages without having to press PF3 for each package. If you activate prompting by selecting option 1A on the Statement Execution Prompt panel (ADB2PSTM), the stacked output will be displayed after the commands are processed.

VERSIONS

Displays version, bind timestamp, and contoken information about the packages in the fifth column. You can issue one of the following variations of the VERSIONS command:

VER ON

Displays the bind timestamp, with version and contoken listed below it.

VER SHORT

Displays only the package version.

VER OFF

Removes package version from display and replaces it with bind timestamp.

VER CON

Removes package version from display and replaces it with contoken.

VER

Cycles between the VER ON, VER SHORT, VER OFF, and VER CON displays each time you issue this command.

GRANT

Issues a GRANT command on multiple application packages.

ALL

Lists all objects of a specified type for each object in a list of objects. Supported values for this panel are:

ALL T Shows all tables for the listed packages.

ALL X Shows all indexes for the listed packages.

PLANMGMT

Displays the plan management attributes for the package. When the PLANMGMT command is used, the panel layout will then include the QUALIFIER command.

DET

Displays the Detail Package report for the selected packages. The following information is displayed for each package:

- Package details
- SQL information
- Explain information from the package owner's plan table

Package details are in one section of the report and the SQL information is in another section of the report.

Explain information is displayed in the SQL information section for each SQL statement that has data in the package owner's plan table.

Note: Explain information for queries that are eligible to be offloaded to a DB2 Analytics Accelerator (accelerator) are also displayed in the SQL information section.

QUALIFIER
Displays the qualifier for the package.

You can also issue the SQ line command to show the SQL statements. These functions are shown at the end of this subsection.

Tip: The BIND, REBIND, FREE, GRANT, PLANMGMT, DET, and QUALIFIER commands operate on each row that is displayed in the table. If you want to omit some of the rows before you issue one of these commands, you can use the minus (-) line command to remove rows from the display. The BIND, REBIND, FREE, GRANT, PLANMGMT, DET, and QUALIFIER commands operate only on rows that are listed.

You cannot BIND a TRIGGER PACKAGE using panel ADB21K. Attempting to do so will result in error ADB272E.

The fields on this panel are:

S Input field where you enter one of the line commands listed on the panel.

COLLECTION
Name of the package collection.

NAME
Name of the package.

OWNER
Authorization ID of the package owner.

CONTOKEN
The consistency token.

BIND TIMESTAMP
Time stamp that indicates when the package was last bound.

VD This field indicates whether validity checking can be deferred until run time. This field contains one of the following values:
  B All validity checking must be done during the bind.
  R Validity checking is done at run time for tables, views, and privileges that do not exist at bind time.

IS Isolation level, which is one of the following values:
  R Repeatable read
  S Cursor stability
  T Read stability
  U Uncommitted read
  Blank Not specified; therefore, at the level specified for the plan

VA This field indicates whether the package is valid, that is, whether it can be run without being rebound. This field contains one of the following values:
  Y Yes
  N No

OP This field indicates whether the package can be allocated. This field contains one of the following values:
  Y Yes

N No. Explicit BIND or REBIND is required before the package can be allocated.

QUALIFIER
Qualifier that was specified at bind time to resolve names.

Plan Mgmt
Plan management attribute of the package.

RL When resources for the package are released. This field contains one of the following values:
C Resources for the package are released at commit time.
D Resources for the package are released at deallocation time.
Blank The value specified for the package is used.

EX This field indicates whether the package was bound using EXPLAIN. This field contains one of the following values:
Y The package was bound using EXPLAIN.
N The package was not bound using EXPLAIN.
Only

EXPLAIN is run. EXPLAIN tables are populated and the BIND process is completed, however, any existing package is not affected.

DR Dynamic SQL rules. This field contains one of the following values:
B Use binder's authid and authorizations.
D DEFINEBIND. When the package is run under an active stored procedure or user-defined function, dynamic SQL statements in the package are executed with DYNAMICRULES define behavior. Otherwise, they are executed with DYNAMICRULES bind behavior.
E DEFINERUN. When the package is run under an active stored procedure or user-defined function, dynamic SQL statements in the package are executed with DYNAMICRULES define behavior. Otherwise, they are executed with DYNAMICRULES run behavior.
H INVOKEBIND. When the package is run under an active stored procedure or user-defined function, dynamic SQL statements in the package are executed with DYNAMICRULES invoke behavior. Otherwise, they are executed with DYNAMICRULES bind behavior.
I INVOKERUN. When the package is run under an active stored procedure or user-defined function, dynamic SQL statements in the package are executed with DYNAMICRULES invoke behavior. Otherwise, they are executed with DYNAMICRULES run behavior.
R Use executor's authid and authorizations.
Blank Not specified. Use the dynamic rules of the plan.

Binding packages

Use the B line command (bind package) on the Packages panel to display the Bind Package panel, as shown in the following figure.

Use the Bind Package panel to build an application package.

Enter your input on the panel.

The following figure shows the Bind Package panel.

```
ADB21KB n --------------------- DB2X  BIND PACKAGE                    13:12
Command ===>

                                                            More:     +
 Verify BIND parameters:

 BIND PACKAGE(
 LOCATION . . . . . .                 >
 COLLECTION . . . . . DSNTIAP              >
 OWNER  . . . . . . . DB2ADM   >
 QUALIFIER  . . . . . DB2ADM   >
 LIBRARY  . . . . . . 'DSN.DBAB.SDSNDBRM'

 MEMBER . . . . . . .                                              >
 SQLERROR . . . . . .          (COntinue, NOpackage or CHeck)
 VALIDATE . . . . . . R        (Run or Bind, Bind preferred)
 ISOLATION  . . . . .          (CS, RR, RS, or UR)
 RELEASE  . . . . . .          (Commit, Deallocate, or blank)
 EXPLAIN  . . . . . .          (Yes, No, or Only)
 CURRENTDATA  . . . . NO       (Yes/No)   (inhibit blocking)
 ACTION . . . . . . . REPLACE  (Add or Replace)
 REPLVER  . . . . . .
                              (replace version)
 ENABLE   . . . . . .          (use ? to get current values from the catalog)
 DISABLE  . . . . . .          (use ? to get current values from the catalog)
 En/disable names . .
                              (use ? to get current values from the catalog)
 DEGREE . . . . . . . 1        (1 or ANY)   (parallelism)
 DYNAMICRULES . . . .          (R, B, D, I, E, H or blank)
 KEEPDYNAMIC  . . . . NO       (Yes/No)
 DEFER(PREPARE)/NO  .          (Yes/No)
 REOPT  . . . . . . . NONE     (N - None, Y - Always, 1 - Once or A - Auto)
 OPTHINT  . . . . . .          >  (hint id)
 PATH (UDT/UDF/STP) .                                              >
 ENCODING . . . . . . 37       (ASCII, EBCDIC, UNICODE, or ccsid)
 IMMEDWRITE . . . . . NO       (Yes, No, or PH1)
 ROUNDING . . . . . . HALFEVEN (Ceiling, Down, Floor, HalfDown,)
                              (HalfEven, HalfUp or Up)
 PLANMGMT . . . . . .          (On or Off)
 PLANMGMTSCOPE  . . .          (S - Static, D - Dynamic, or A - All)
 APREUSE  . . . . . .          (Yes/No)
 APCOMPARE  . . . . .          (N - None, W - Warn, or E - Error)
 BUSTIMESENSITIVE . . YES      (Yes/No)
 SYSTIMESENSITIVE . . YES      (Yes/No)
 APPLCOMPAT . . . . .          (V10R1/V11R1)
 EXTENDEDINDICATOR  .          (Yes/No)
 CONCURRENTACCESSRES           (U — Usecurrentlycommitted or)
                              (W - Waitforoutcome)

 )
```

Figure 557. Bind Package panel (ADB21KB)

Rebinding packages

Use the RB line command (rebind package) on the Packages panel to display the
Rebind Package panel, as shown in the following figure.

Use the Rebind Package panel to rebind an application package when changes
have been made that affect the package, but the SQL statements in the program
have not changed.

The PLANMGMT option should be OFF or BLANK when a REBIND of a package
is changed to a different OWNER or QUALIFIER. For example, when the OWNER
is changed from SYSADM to ADMF001.

```
ADB21KR n -------------------- DB2X Rebind Package -------------------- 13:20
Command ===>

                                                         More:    +
Verify REBIND parameters:

REBIND PACKAGE(
Location . . . . . .             >
Collection . . . . . DSNEDCL          >
Package  . . . . . . DSNECP68 >
(
Version  . . . . . . V10R1

OWNER  . . . . . . . DB2ADM   >
QUALIFIER  . . . . . DB2ADM   >
VALIDATE . . . . . . R        (Run or Bind, Bind preferred)
ISOLATION  . . . . . CS       (CS, RR, RS, or UR)
RELEASE  . . . . . .          (Commit, Deallocate, or blank)
EXPLAIN  . . . . . .          (Yes, No, or Only)
CURRENTDATA  . . . . YES      (Yes/No)   (inhibit blocking)
ENABLE   . . . . . .          (use ? to get current values from the catalog)
DISABLE  . . . . . .          (use ? to get current values from the catalog)
En/disable names . .
                              (use ? to get current values from the catalog)
DEGREE . . . . . . . 1        (1 or ANY) (parallelism)
DYNAMICRULES . . . .          (R, B, D, I, E, H or blank)
KEEPDYNAMIC  . . . . NO       (Yes/No)
DEFER(PREPARE) . . .          (Yes/No)
REOPT  . . . . . . . NONE     (N - None, Y - Always, 1 - Once, A - Auto)

OPTHINT  . . . . . .          >  (hint id)
PATH (UDT/UDF/STP) .                           >
ENCODING . . . . . . 37       (ASCII, EBCDIC, UNICODE, or ccsid)
IMMEDWRITE . . . . . NO       (Yes, No, or PH1)
PLANMGMT . . . . . .          (On, Off, Basic or Extended)
SWITCH . . . . . . .          (Original or Previous - ALL OTHER OPTIONS IGNORE
ROUNDING . . . . . . HALFEVEN (Ceiling, Down, Floor, HalfDown,)
                              (HalfEven, HalfUp or Up)
PLANMGMTSCOPE  . . .          (S - Static, D - Dynamic or A - All)
APREUSE  . . . . . .          (Yes/No)
APCOMPARE  . . . . .          (N - None, W - Warn, E - Error)
BUSTIMESENSITIVE . . YES      (Yes/No)
SYSTIMESENSITIVE . . YES      (Yes/No)
APRETAINDUP  . . . .          (Yes/No)
EXTENDEDINDICATOR  .          (Yes/No)
CONCURRENTACCESSRES           (U – Usecurrentlycommitted or)
                              (W - Waitforoutcome)
)
```

Figure 558. Rebind Package panel (ADB21KR)

Freeing packages

Use the F line command (free package) on the Packages panel to display the Free
Package panel, as shown in the following figure.

Use the Free Package panel to delete a specific version of a package, all versions of
a package, or whole collections of packages.

Enter your input on the panel.

```
B21KF n --------------------- DB2X Free Package --------------------- 03:28
 Command ===> _____


  FREE PACKAGE (

  Location  . . .  _____  >  (Blank for local)

  Collection  . . ADBLTJ              >

  Name  . . . . . ADB2REP  >

  (
  Version . . . .  _____

 )) PLANMGMTSCOPE(
  Scope . . . . .  _____             (All, Inactive)
  )
```

Figure 559. Free Package panel (ADB21KF)

CAUTION:
**If you specify an asterisk (*) for collection, all packages with the specified name
and version number are freed. If you specify a collection name and an * for both
Name and Version, all packages in that collection are freed. Thus, the use of
asterisks can be very powerful, and should be used carefully.**

Displaying detailed package information

Use the DET line command on the Packages panel to display the Details for
object(s) panel. The following figure shows the Package details with the SQL
information section collapsed.

```
ADBPD                  DSNA Details for object(s)
Command ===>                                            Scroll ===> PAGE

Commands: SAVE  ZOOM

_ Details for package : SPADJB009012345678901(*1) in collection : SCADJB009(*2)

_  Package information
     Package type . . . . . . . . . . : Native SQL routine package
     Version  . . . . . . . . . . . . : MYVERSION
     Authorization ID of owner  . . . : J148286
     Owner type . . . . . . . . . . . : Auth ID
     Authorization ID of creator  . . : VNDR001
     Created timestamp  . . . . . . . : 2012-08-23-05.38.20.906062
     Latest BIND timestamp  . . . . . : 2012-11-06-16.42.39.648458
     Version under which package bound: V11
     Qualifier for unqualified SQL  . : J148286
     Operative status of package  . . : Package is valid and operative
     Resource and authorization check : At BIND time
     Size of the base section (bytes) : 4272    (in EDM pool during execution)
     Average DML section size (bytes) : 5220    (loaded when needed during exec)
     Package bound with EXPLAIN . . . : Yes
     SQLERROR specified at BIND time  : No - SQLERROR(NOPACKAGE) specified
     BIND or REBIND from remote loc.  : No - (RE)BIND was from a local system
     Remote packages creation method  :
     Source of the package  . . . . . :
     Number of enabled/disabled conn. : 0
     Data concurrency . . . . . . . . : B - not required
       Effect on blocking . . . . . . : Allow blocking for ambiguous cursors
     DEGREE of I/O parallelism  . . . : 1 - parallel I/O inhibited
     Group member that performed BIND :
     Dynamic SQL rules  . . . . . . . : Use definers authid and authorizations
     Re-optimize SQL at execution time: 1 - use exec. time variable values once
     Defer prepare  . . . . . . . . . : Yes - prepare is defered to OPEN time
     Keep prepared dynamic SQL stmts  : No - are destroyed at COMMIT
     Protocol for 3 part names  . . . : D - uses DRDA
     Function resolved at . . . . . . : 2012-11-06-16.42.39.648445
     Optimizer hint identifier  . . . : THIS IS THE OPTHINT FOR JB
     Encode CCSID . . . . . . . . . . : 37
     Write group buffer pool pages  . : Immediate write
     ROUNDING option used on last bind: Round Down
     Concurrent Access  . . . . . . . : W - Wait for release of write lock
     SQL path for resolving UDT,UDF,SP: "J148286","SYSADM","USRT001"

     Precompiler related information:
       Timestamp of precompilation  . : 0001-01-01-00.00.00.000000
       Consistency token in hex . . . : 1941FCD60BBACC4D
       SQL escape character . . . . . : ' (apostrophe)
       Decimal point character  . . . : . (period)
       Host program language  . . . . : Remotely bound, trigger, or SQL package
       Mixed character set  . . . . . : N
       Decimal 31 used  . . . . . . . : Yes
       Katakana . . . . . . . . . . . : No

     Resource allocation information:
       Resources are released . . . . : At plan deallocation time
       Isolation level  . . . . . . . : Read stability

_  SQL statements in package: SCADJB00901234567890.SPADJB009012345678901234(*3)
  _____

_ Long names legend

  (*1) - SPADJB00901234567890123456E
  (*2) - SCADJB00901234567890
  (*3) - SCADJB00901234567890.SPADJB00901234567890123456E.MYVERSION
```

Figure 560. Details for object(s) (ADBPD)

The following figure shows the SQL information section with the Package
information section collapsed.

```
ADBPD                   DSNA Details for object(s)
Command ===>                                               Scroll ===> PAGE

Commands: SAVE  ZOOM

_ Details for package : SPADJB009012345678901(*1) in collection : SCADJB009(*2)

_ Package information
_ SQL statements in package: SCADJB00901234567890.SPADJB009012345678901234(*3)

_ SQL in statement: 39
_ Explain information for SQL statement: 39
_ SQL in statement: 39
_ SQL in statement: 40
_ SQL in statement: 42
    INSERT INTO SCADJB00.TBADJB00 (ORDER_WAREHOUSE_ID) VALUES ('EEE')

_ Explain information for SQL statement: 42

   The operation is INSERT, UPDATE or DELETE.
   Inner join or no join.
   --------------------------------------------------------------------------
   Table Schema . . . : SCADJB00      Table Name . . . . : TBADJB00
   Query number . . . : 42           Access type  . . . :
   Plan number  . . . : 0            Query block no . . : 1
   Match columns  . . : 0

_ SQL in statement: 39
  CLOSE
   C1

_____

_ Long names legend

  (*1) - SPADJB00901234567890123456E
  (*2) - SCADJB00901234567890
  (*3) - SCADJB00901234567890.SPADJB00901234567890123456E.MYVERSION
```

Figure 561. Details for object(s) (ADBPD)

SQL statements are presented in 72 byte line lengths. If a statement contains host variables, the variable name and data type are displayed on separate, new lines.

Note: If the package contains a query that is marked to be offloaded to an accelerator, a figure similar to the following is displayed instead. Accelerated queries have an access type of A (accesstype = 'A').

```
ADBPD min ---------------- DSNB Details for object(s) --------------- 15:55
Command ===>                                             Scroll ===> PAGE

Commands: SAVE  ZOOM

_ Details for package : ADM1PK01              in collection : RRLCOL

_  Package information
_  SQL statements in package: RRLCOL.ADM1PK01

_ SQL in statement: 1686
      SELECT * INTO
        :policyid                           Var Char(10) ,
        :coverage                           Integer ,
        :start                              Var Char(49) ,
        :COUNT                              Integer ,
        :timeid                             Var Char(49)
      FROM SCADM101.TBADM101

_   Explain information for SQL statement: 1686

     Query is marked to be offloaded to an accelerator.
     Query qualifies for routing to an accelerator.
     ----------------------------------------------------------------------
     Table schema . . . : SCADM101     Table name . . . . : TBADM101
     Query blk no . . . : 1            Access type  . . . : A
     Accelerator name . : ZGRYPHON     Location name  . . : DB2EC1
     Reason code  . . . : 0

_____

****************************** Bottom of data ********************************
```

Figure 562. Details for object(s) (ADBPD)

Viewing extracted SQL for a package

Use the SQ line command (show SQL) on the Packages panel to display the
Extracted SQL panel, as shown in the following figure.

The Extracted SQL panel displays the SQL statements in a package.

```
ADB21KSE  ----------------- Extracted SQL ---------------- Columns 00001 00072
Command ===>                                             Scroll ===> CSR

****** **************************** Top of Data *****************************
==MSG> Use primary command "EXPLAIN" to explain or PLANTAB to display the
==MSG> explain rows for the selected SQL statement using line command "C" or
==MSG> block line command "CC".
==MSG>
=NOTE= -- SQL statements in  PACKAGE : ADBB2PAR.ADB2REM.(V11.2.0.0000000)
=NOTE= -- SQL in stmt:    3041 (Stmt id:589559)
000001 SET :H = GETVARIABLE ('SYSIBM.PLAN_NAME', 'D_PLNAME' )
=NOTE= -- SQL in stmt:    3048 (Stmt id:589560)
000002 SET :H = GETVARIABLE ('SYSIBM.PACKAGE_SCHEMA', 'D_PKSCH' )
 ****** **************************** Bottom of Data ****************************
```

Figure 563. Extracted SQL panel (ADB21KSE)

The following primary commands are valid on this panel:

EXPLAIN
> Explains the selected SQL statement when you issue the C line command or
> the CC block line command. Navigates to EXPLAIN panel ADB2EL.

PLANTAB
Displays explain rows for the selected SQL statement when you issue the C
line command or the CC block line command. Navigates to EXPLAIN panel
ADB2EL

Option L. Collections

The Collections panel displays the collections in the DB2 catalog.

A *collection* is a group of associated packages. Binding packages into package
collections allows you to add packages to an existing application plan without
having to bind the entire plan again.

Displaying collections

Select option L on the System Catalog panel to display the Collections panel, as
shown in the following figure.

On the Collections panel, you can issue the SQ line command to show the SQL
statements. This function is shown in "Viewing extracted SQL for a package in a
collection" on page 1001.

The following figure shows the Collections panel.

```
 DB2 Admin ------------------ DB2X Collections -------------------- Row 1 of 27
 Command ===>                                            Scroll ===> PAGE

 Line commands:
  K - Packages in collection  PL - Package lists  P - Local plans
  A - Authorizations  GR - Grant  SQ - SQL in packages in collection

                          Number of
 S     Collection         Packages
       *                       *
 ---- ------------------> -----------
       ADBL                   6
       ADBL21                 11
       ADBL31                 7
       ADBV3                  3
       ADB21                  1
       DSNEDCL                1
       DSNESPCS               1
       DSNESPRR               1
       DSNHYCRDRDRDABRAGG     1
       DSNREXCS               1
       DSNREXRR               1
       DSNREXRS               1
       DSNREXUR               1
       DSNREXX                1
       DSNTEP2                1
 ****************************** END OF DB2 DATA ********************************
```

Figure 564. Collections panel (ADB21L)

The fields on this panel are:

S   Input field where you enter one of the line commands listed on the panel.

Collection
      Name of the package collection.

Number of Packages
      Number of packages in the collection.

Viewing extracted SQL for a package in a collection

The Extracted SQL panel, as shown in the following figure, is displayed when you issue line command SQ (show SQL) on the Collections panel.

This panel displays the SQL statements in a package shown on the Collections panel.

The following figure shows the Extracted SQL panel.

```
 DB2 Admin ----------------- Extracted SQL ---------------- Columns 00001 00072
 Command ===>                                               Scroll ===> PAGE
 Max no of rows reached
 ****** ****************************** Top of Data ******************************
 000001 -- SQL statements in PACKAGE : ADBL31.ADBMAIN.()
 000002 -- SQL in stmt: 605
 000003 COMMIT WORK
 000004 -- SQL in stmt: 2601
 000005 DECLARE S1 STATEMENT
 000006 -- SQL in stmt: 2643
 000007 PREPARE S1 FROM :H
 000008 -- SQL in stmt: 2747
 000009 DESCRIBE S1 INTO :H
 000010 -- SQL in stmt: 2759
 000011 EXECUTE S1
 000012 -- SQL in stmt: 2884
 000013 DECLARE C1 CURSOR FOR S1
 000014 -- SQL in stmt: 2890
 000015 OPEN C1
 000016 -- SQL in stmt: 2902
 000017 FETCH C1 USING DESCRIPTOR :H
 000018 -- SQL in stmt: 2973
 000019 CLOSE C1
 000020 -- SQL in stmt: 5754
 000021 COMMIT WORK
 000022 -- SQL in stmt: 5781
 000023 ROLLBACK WORK
 000024 -- SQL in stmt: 5786
 000025 COMMIT WORK
 000026 -- SQL statements in PACKAGE : ADBL31.ADB2CON.()
 000027 -- SQL in stmt: 123
 000028 CONNECT RESET
 000029 -- SQL in stmt: 128
 000030 CONNECT
 000031 -- SQL in stmt: 134
 000032 CONNECT TO :H
 000033 -- SQL statements in PACKAGE : ADBL31.ADB2GEN.()
 000034 -- SQL in stmt: 1917
 000035 DECLARE C_SYSDAUTH CURSOR FOR SELECT * FROM SYSDBAUTH WHERE NAME = :H
 000036 AND GRANTOR <> GRANTEE ORDER BY DATEGRANTED, TIMEGRANTED
 000037 -- SQL in stmt: 1931
 000038 OPEN C_SYSDAUTH
```

Figure 565. Extracted SQL panel (ADB21KSE)

Option N. Constraints

The Constraints panel displays the constraints on a table in the DB2 catalog.

Select option N on the System Catalog panel to display the Constraints panel, as shown in the following figure.

```
DB2 Admin ------------------ DB2X Constraints ---------- Row 1 to 12 of 1,000
Command ===>                                         Scroll ===> PAGE
Max no of rows reached
Line commands:
 S - Show  T - Table

    Table
Sel Schema   Table Name        Constraint Name                                Type
    *        *                 *                                              *
---- -------- ------------------ --------------------------------------------> ----
    SYSIBM   SYSINDEXPART      IXCREATOR                                      U
    SYSIBM   SYSINDEXSTATS     OWNER                                          U
    SYSIBM   SYSJAROBJECTS     JARSCHEMA                                      P
    SYSIBM   SYSLOBSTATS       DBNAME                                         P
    SYSIBM   SYSTABCONST       TBCREATOR                                      P
    SYSIBM   SYSTABLEPART      DBNAME                                         U
    SYSIBM   SYSTABLESPACE     DBNAME                                         P
    SYSIBM   SYSTABSTATS       OWNER                                          U
    VNDX01   EMP_PHOTO_RESUME  EMPNO                                          P
    I2MADMIN ICMUT00302001     COMPKEY                                        P
    VNDOXL2  DEPT              DEPTNO                                         P
    I2MADMIN ICMSTITEMSTODELETE ITEMID                                        P
```

Figure 566. Constraints panel (ADB21N) – partial display

The fields on this panel are:

Sel
>Enter one of the line commands listed on the panel.

Table Schema
>The schema of the table on which the constraint is defined.

Table Name
>The name of the table.

Constraint Name
>The name of the constraint.

Type
>The type of constraint. Possible values are:
>
>**P**   Primary key
>
>**U**   Unique
>
>**F**   Foreign key

Option O. Stored Procedures

Use the Stored Procedures panel to display information about the stored procedures in the DB2 catalog.

Select option O on the System Catalog panel to display the Stored Procedures panel, as shown in the following figure.

```
 ADB210 in ---------------- DSNB Stored Procedures ---------- Row 1 to 9 of 363

 Commands: GRANT
 Line commands:
  AH - Schema Auth  A - Auth  DROP - Drop  AL - Alter  K - Package  PA - Parms
  DIS - Display  STO - Stop  STA - Start  GR - Grant  COM - Comment  CALL - Call
  ? - Show all line commands
                                                           S
                                                   Res   Q S P C External
 Sel  Schema    Name                Version  A Lang Parms  Set O L R T R Name
      *         *                   *        * *         *   * * * * * * *
 ---- -------- ------------------- -------- - ---- ------ --- - - - - - - ---------
      SYSPROC  ADMIN_COMMAND_DB2            C      12   2 E M N M N DSNADMCD
      SYSPROC  ADMIN_COMMAND_DSN           REXX    2   1 E M N M N DSNADMCS
      SYSPROC  ADMIN_COMMAND_MVS           C      11   1 E M N M N DSNADMCM
      SYSPROC  ADMIN_COMMAND_UNIX          C       6   1 E M N M N DSNADMCU
      SYSPROC  ADMIN_DS_BROWSE             ASSE    6   1 E M N M N DSNADMDB
      SYSPROC  ADMIN_DS_DELETE             ASSE    6   0 E M N M N DSNADMDD
      SYSPROC  ADMIN_DS_LIST               ASSE    7   1 E M N M N DSNADMDL
      SYSPROC  ADMIN_DS_RENAME             ASSE    7   0 E M N M N DSNADMDR
      SYSPROC  ADMIN_DS_SEARCH             ASSE    6   0 E M N M N DSNADMDE
```

Figure 567. Stored Procedures panel (ADB21O)

The following primary command is valid on this panel:

GRANT
Issues a GRANT command on multiple stored procedures.

Tip: The GRANT command operates on each row that is displayed in the
table. If you want to omit some of the rows before you issue the command,
you can use the minus (-) line command to remove rows from the display. The
GRANT command operates only on rows that are listed.

The fields on this panel are:

SEL
Input field where you enter one of the line commands listed on the panel.

SCHEMA
Schema of the stored procedure.

NAME
Name of the stored procedure.

VERSION
The version identifier for a native SQL procedure.

A Identifies the active version of a routine.

LANG
Implementation language.

PARMS
Number of parameters for the stored procedure.

RES SET
Maximum number of result sets that can be returned.

O Origin of the routine:

> **E**   External.
>
> **Q**   SQL.
>
> **S**   System generated.
>
> **U**   User-defined or built-in function.

N   Native SQL procedure.

SQL

This field indicates whether SQL statements are allowed, which is one of the
following values:

N   No SQL statement, SQL is not allowed.
C   Contains SQL statements.
R   Reads SQL data.
M   Modifies SQL data.
blank Not applicable.

SR This field indicates whether the program should remain resident when it ends.
This field contains one of the following values:
Y   Program remains resident.
N   Program does not remain resident.
blank Not external or user-defined function.

PT Program type, which is one of the following values:
M   Main.
S   Subroutine.
blank Not applicable.

CR Commit on return. This field contains one of the following values:
Y   Unit of work is committed immediately.
N   Unit of work continues.
A   Autonomous. Only the unit of work from the procedure is committed.
Work from the application that calls the procedure is not immediately
committed.

EXTERNAL NAME

Load module name for the stored procedure.

Option P. Plans

The Plans panel displays the application plans in the DB2 catalog.

Select option P on the System Catalog panel to display the Application Plans panel,
as shown in the following figure.

By using the Application Plans panel, you can issue line commands to bind,
rebind, and free an application plan. These functions are shown at the end of this
subsection. You can also issue the SQ line command to show the SQL statements.
The SQ line command applies to all packages in a plan and therefore can affect
performance.

The following figure shows the Application Plans panel.

```
DB2 Admin --------------- DB2X Application Plans --------------- Row 1 of 25
Command ===>                                                Scroll ===> PAGE

Commands: BIND  REBIND  FREE  GRANT
Line commands:
 DP - Depend  A - Auth  T - Tables  V - Views  X - Indexes  S - Table spaces
 Y - Synonyms  M - DBRMs  RB - Rebind  F - Free  B - Bind  GR - Grant
 PL - Package list  LP - List PLAN_TABLE  I - Interpret  ENDI - Enab/disab con
 K - Local packages  SQ - SQL  D - Databases  RO - role
                         Bind   Bind   V I V O Bound    Quali-     Pack A R E D
Select Name     Owner    Date   Time   D S A P By       fier       Lists Q L X R
       *        *        *      *      * * * * *        *                * * * * *
------ -------- -------> ------ ------ - - - - -------- -------- ------ - - - -
       ADBTEP2  DSCGDB2  010828 100153 B S Y Y ISTFL2   DSCGDB2      1 U C N
       ADBV3    DSCGDB2  010912 024459 B S Y Y ISTFL    DSCGDB2      2 U C Y
       ADB2GEN  DSCGDB2  010623 005531 B S Y Y ISTJE    DSCGDB2      1 U C Y
       ADB2GE2  DSCGDB2  010526 003803 B S Y Y ISTFL    DSCGDB2      1 U C Y
       ADB21    DSCGDB2  010623 004026 B S Y Y ISTJE    DSCGDB2      1 U C N
       ADB31    DSCGDB2  011030 170150 B S Y Y ISTJE    DSCGDB2      1 U C N
       DB2E71   DPGROTH  011029 145636 R S Y Y DPGROTH  DPGROTH      0 U C Y
       DSNEDCL  DSCGDB2  010524 190326 R S Y Y ISTJE    DSCGDB2      1 U C N
       DSNESPCS DSCGDB2  010524 190324 R S Y Y ISTJE    DSCGDB2      1 U C N
       DSNESPRR DSCGDB2  010524 190325 R R Y Y ISTJE    DSCGDB2      1 U C N
       DSNHSP81 ISTJE001 010524 202509 R S Y Y ISTJE    ISTJE        0 U C N
       DSNHYCRD DSCGDB2  010524 190331 R S Y Y ISTJE    DSCGDB2      1 U C N
       DSNREXX  DSCGDB2  010524 190846 R S Y Y ISTJE    DSCGDB2      5 U C N
       DSNTEP2  DSCGDB2  010524 202123 R S Y Y ISTJE    DSCGDB2      1 U C N
       DSNTEP81 DSCGDB2  010524 202123 R S Y Y ISTJE    DSCGDB2      1 U C N
       DSNTIAD  DSCGDB2  010524 024119 R S Y Y ISTJE    DSCGDB2      0 U C N
       DSNTIA81 DSCGDB2  010524 024119 R S Y Y ISTJE    DSCGDB2      0 U C N
       DSNTIB81 DSCGDB2  010525 033553 R S Y Y ISTJE    DSCGDB2      0 U C N
       DSNWZP   DSCGDB2  010524 190331 R S Y Y ISTJE    DSCGDB2      1 U C N
       DSN8EPU  DSCGDB2  010601 204822 R S Y Y ISTJE    DSCGDB2      2 U C N
       GOC2GEN  DSCGDB2  010829 100859 B S Y Y ISTFL    DSCGDB2      1 U C Y
       SKALBERG DPCHR    010622 143748 R U Y Y DPCHR    DPCHR        3 U D Y
       TADB2RE  DSCGDB2  011022 162840 R R Y Y ISTFL    DSCGDB2      2 U C N
       TESTPRP  ISTFL    010526 004951 B S Y Y ISTFL    ISTFL        1 U C N
****************************** END OF DB2 DATA ******************************
```

Figure 568. Application Plans panel (ADB21P)

The following primary commands are valid on this panel:

BIND

Issues a BIND command on multiple application plans. When you attempt to bind more than 20 application plans, DB2 Admin prompts you to specify either a work statement list or a batch job to complete the processing.

FREE

Issues a FREE command on multiple application plans. When you attempt to free more than 20 application plans, DB2 Admin prompts you to specify either a work statement list or a batch job to complete the processing.

REBIND

Issues a REBIND command on multiple application plans. When you attempt to rebind more than 20 application plans, DB2 Admin prompts you to specify either a work statement list or a batch job to complete the processing.

When you specify REBIND, the resulting BIND command contains only the plan name. Specify REBIND FULL. If you want the resulting BIND command to contain the plan name and all of the parameters.

GRANT

Issues a GRANT command on multiple application plans.

Tip: The BIND, REBIND, FREE, and GRANT commands operate on each row that is displayed in the table. If you want to omit some of the rows before you issue

one of these commands, you can use the minus (-) line command to remove rows from the display. The BIND, REBIND, FREE, and GRANT commands operate only on rows that are listed.

The fields on this panel are:

Select
Input field where you enter one of the line commands listed on the panel.

Name
Name of the application plan.

Owner
Authorization ID of the owner of the application plan.

Bind Date
Date of the most recent bind on the application plan. The date is in the form YYMMDD.

Bind Time
Time of the most recent bind on the application plan. The time is in the form HHMMSS.

VD This field indicates whether validity checking can be deferred until run time. This field contains one of the following values:
    B   All validity checking must be done during the bind.
    R   Validity checking is done at run time for tables, views, and privileges that do not exist at bind time.

IS Isolation level, which is one of the following values:
    R   Repeatable read
    S   Cursor stability
    T   Read stability
    U   Uncommitted read

VA This field indicates whether the application plan is valid; that is, whether it can be run without being rebound. This field contains one of the following values:
    Y   A valid application plan.
    N   Not a valid application plan.
    A   The description changed. The application plan is still valid.
    H   The description changed. The application plan is valid for DB2 Version 5 or higher; otherwise, the plan is invalid.

OP This field indicates whether the application plan can be allocated. This field contains one of the following values:
    Y   Yes
    N   No. Explicit BIND or REBIND is required before the plan can be allocated.

Bound By
Primary authorization ID of the binder of the plan.

Qualifier
Qualifier that was specified at bind time to resolve names.

Pack Lists
Number of packages in the package list at bind time.

AQ When resources for the application plan are acquired. This field contains one of the following values:
    A   At allocation time
    U   At first use

RL When resources for the application plan are released. This field contains one of the following values:

C   Resources for the application plan are released at commit time.

D   Resources for the application plan are released at deallocation time.

EX This field indicates whether the application plan was bound using EXPLAIN. This field contains one of the following values:

Y   Yes

N   No

DR Dynamic SQL rules. This field contains one of the following values:

B   Use binder's authid and authorizations.

Blank Use executor's authid and authorizations.

Binding application plans

Use the B line command (bind plan) on the Application Plans panel to display the Bind Application Plan panel, as shown in the following figure.

Use the Bind Application Plan panel to build an application plan.

Enter your input on the panel.

The following figure shows the Bind Application Plan panel.

```
ADB21PB n ----------------- DB2X Bind Application Plan ----------------- 13:41
Command ===>

                                                             More:    +
 Verify BIND parameters:

 BIND PLAN(
 Plan name  . . . . . DSNESPRR
 OWNER  . . . . . . . DB2ADM    >
 QUALIFIER  . . . . . DB2ADM    >  (qualifier to resolve unqualified SQL)
 PKLIST . . . . . . . *.DSNESPRR.DSNESM68 *.DSNTIAP.DSNTIAP        >
 DEFER(PREPARE) . . . NO       (Yes/No, used for distributed dynamic SQL)
 VALIDATE . . . . . . R        (Run or Bind, Bind preferred)
 ISOLATION  . . . . . RR       (CS, RR, RS, or UR)
 CACHE  . . . . . . . 3072     (cache size in bytes for authorization IDs)
 ACQUIRE  . . . . . . U        (Use or Allocate, Use preferred)
 RELEASE  . . . . . . C        (Commit or Deallocate, Commit preferred)
 EXPLAIN  . . . . . . NO       (Yes/No, to explain access path)
 CURRENTDATA  . . . . NO       (Yes/No)
 CURRENT SERVER . . .               >  (blank=local, else first location)
 ACTION . . . . . . . REPLACE  (Add or Replace)
 RETAIN . . . . . . . YES      (Yes/No)  (Retain auth list)
 ENABLE . . . . . . .          (use ? to get current values from the catalog)
 DISABLE  . . . . . .          (use ? to get current values from the catalog)
 En/disable names . .
                               (use ? to get current values from the catalog)
 DEGREE . . . . . . . 1        (1 or ANY)  (Parallelism)
 SQLRULES . . . . . . D        (DB2 or STD)
 DISCONNECT . . . . . E        (Explicit, Automatic, or Conditional)
 DYNAMICRULES . . . .          (Run or Bind)
 KEEPDYNAMIC  . . . . NO       (Yes/No)
 REOPT(VAR) . . . . . NONE     (N - None, Y - Always, 1 - Once, or A-Auto)
 OPTHINT  . . . . . .          >
 PATH . . . . . . . .                                            >
 ENCODING . . . . . . 37       (ASCII, EBCDIC, UNICODE or ccsid)
 IMMEDWRITE . . . . . NO       (Yes,No or PH1)
 ROUNDING . . . . . . HALFEVEN  (Ceiling, Down, Floor, HalfDown,)
                               (HalfEven, HalfUp, or Up)
 CONCURRENTACCESSRES           (U - Usecurrentlycommitted or)
                               (W - Waitforoutcome)
```

Figure 569. Bind Application Plan panel (ADB21PB)

Rebinding application plans

Use the RB line command (rebind plan) on the Application Plans panel to display
the Rebind Application Plan panel, as shown in the following figure.

Use the Rebind Application Plan panel to rebind an application plan when changes
have been made that affect the plan, but the SQL statements in the program have
not changed.

Enter your input on the panel.

```
ADB21PR n ---------------- DB2X Rebind Application Plan ---------------- 13:48
Command ===>

                                                          More:     +
 Verify REBIND parameters:

 REBIND PLAN(
 Plan name  . . . . . ADB27AC
 OWNER  . . . . . . . J148286  >
 QUALIFIER  . . . . . J148286  >  (qualifier to resolve unqualified SQL)
 PKLIST . . . . . . .                                              >
 NOPKLIST . . . . . .           (Yes/No, to remove current package list)
 DEFER(PREPARE) . . . NO        (Yes/No, used for distributed dynamic SQL)
 VALIDATE . . . . . . B         (Run or Bind, Bind preferred)
 ISOLATION  . . . . . CS        (CS, RR, RS, or UR)
 CACHE  . . . . . . . 3072      (cache size in bytes for authorization IDs)
 ACQUIRE  . . . . . . U         (Use or Allocate, Use preferred)
 RELEASE  . . . . . . C         (Commit or Deallocate, Commit preferred)
 EXPLAIN  . . . . . . NO        (Yes/No, to explain access path)
 CURRENTDATA  . . . . NO        (Yes/No) (Inhibit blocking)
 CURRENT SERVER . . .               >  (blank=local, else first location)
 ENABLE . . . . . . .           (use ? to get current values from the catalog)
 DISABLE  . . . . . .           (use ? to get current values from the catalog)
 En/disable names . .           (use ? to get current values from the catalog)

 DEGREE . . . . . . . 1         (1 or Any)  (Parallelism)
 SQLRULES . . . . . . D         (DB2 or STD)
 DISCONNECT . . . . . E         (Explicit, Automatic, or Conditional)
 DYNAMICRULES . . . .           (Run or Bind)
 KEEPDYNAMIC  . . . . NO        (Yes/No)
 REOPT(VAR) . . . . . NONE      (N - None, Y - Always, 1 - Once or A - Auto)
 OPTHIN . . . . . . .         >  (hint id)
 PATH . . . . . . . .                                              >
 ENCODING . . . . . . 37        (ASCII, EBCDIC, UNICODE or ccsid)
 IMMEDWRITE . . . . . NO        (Yes, No or PH1)
 ROUNDING . . . . . . HALFEVEN  (Ceiling, Down, Floor, HalfDown,)
                                (HalfEven, HalfUp or Up)
 CONCURRENTACCESSRES            (U — Usecurrentlycommitted or)
                                (W - Waitforoutcome)
)
```

Figure 570. Rebind Application Plan panel (ADB21PR)

Freeing application plans

Use the F line command (free plan) on the Application Plans panel (see Figure 568 on page 1005) to display the Free Application Plan panel, as shown in the following figure.

Use the Free Application Plan panel to delete application plans from DB2.

```
DB2 Admin ------------------ DB2X Free Application Plan ----------------- 01:12
Command ===>


  FREE PLAN

  Plan name ===> DSNTIA81
```

Figure 571. Free Application Plan panel (ADB21PF)

Option PDC. DB2 Pending Definition Changes

Use the DB2 Pending Definition Changes panel to display information about the definition changes that are pending in the DB2 catalog.

Select option PDC on the System Catalog panel to display the DB2 Pending Definition Changes panel, as shown in the following figure.

```
ADBPPDC n --------- DSNB DB2 Pending Definition Changes----- Row 1 to 10 of 64

Commands:  DIS   UTIL   DROP
Line commands:
 T - Tables  D - Database  X - Indexes  S - Table spaces  UTIL - Utilities
 DIS - Display object  DROP - Drop changes  SQ - Statement text  I -Interpret
 ? - Show all line commands

Sel   Name               Qual     T  Seqno Keyword     Value     Timestamp
      *                  *        * *      *           *         *
----- -----------------> -------> - ------ ----------- --------> ------------>
      EMP                T4389Z   T      1 ENDING AT   ('000025' 2013-06-19-23
      PJMQT3             CH86386  T      1 ENDING AT   (12)      2013-05-08-14
      PJMQT4             MA65210  T      1 ENDING AT   (12)      2013-05-08-14
      PJTBP              MKZ1045  T      1 ENDING AT   (11)      2013-05-08-10
      PJTBPDT            SMITH01  T      1 RESTRICT              2013-05-07-09
      PSVTBA01_MAXLEN012 PSVSCHA0 T      1 ENDING AT   (1900,'AA 2013-09-24-15
      PSVTBA02_MQT_MAXLE PSVSCHA0 T      1 ENDING AT   (1900,'AA 2013-09-24-15
      PSVTBA02_MQT_MAXLE PSVSCHA0 T      1 ENDING AT   (3900,'CC 2013-09-24-15
      T4_MQT             S29635_T T      1 ENDING AT   (300,400) 2013-06-28-08
      T4_MQT             S29635_T T      1 ENDING AT   (350,450) 2013-06-28-09
```

Figure 572. DB2 Pending Definition Changes panel (ADBPPDC)

The following primary commands are valid on this panel:

DIS
Performs DB2 DISPLAY command on the listed objects.

UTIL
Generates a utility JCL for all table spaces.

DROP
Drops the pending DB2 changes that are listed.

The following fields are displayed on this panel:

Select
Input field where you enter one of the line commands listed on the panel.

Name
Name of the object that has pending changes.

Qualifier
For a table space, the qualifier is the database name. For an index or table, the qualifier is the schema name.

T Type of object, which is one of the following values:
  S Table space
  I Index
  T Table

Keyword
The keyword of a pending change.

Value
This field shows the text of the value in the pending change.

If the text is truncated, type EXPAND on the primary command line, position the cursor on the default text field, and press Enter to display all of the text.

> **Timestamp**
>> This field indicates when then the pending change was created.

Option Q. Sequences

The Sequence Objects panel displays the sequences in the DB2 catalog.

A *sequence* is a user-defined object that generates a sequence of numeric values according to the specification with which the sequence was created. It efficiently provides recoverable, guaranteed-unique, sequential numbers to DB2 applications.

Select option Q on the System Catalog panel to display the Sequence Objects panel, as shown in the following figure.

On the Sequence Objects panel, you can issue the GEN primary command to generate SQL from DB2 catalog for all displayed sequences. You can also issue the GRANT primary command to change authorizations for all displayed sequences.

```
  DB2 Admin ---------------- DB2X Sequence Objects ---------- Row 1 to 13 of 148
  Command ===>                                               Scroll ===> PAGE

  Commands:  GRANT
   Line commands:
   A - Auth  CRE - Create  AL - Alter  GR - Grant  DROP - Drop  DDL - Object DDL
   IDC - Identity columns  GEN - Generate DDL  F - Functions  J - Triggers
   ALIAS - Alias ? - Show all line commands
  Sel  Schema    Name               Owner     T C                   Start value
       *         *                  *         * *                             *
  ---- -------- ------------------ -------- - - -------------------------------
       ISTJE12  SEQCXM2PPZSOTH8    ISTJE12  A N                             500
       K351156  SEQCXM276GG9TUE    K351156  I Y                               1
       ISTJE10  SEQCXN7K6P3NXDR    ISTJE10  I N                               1
       VNDSHL2  SEQ13              ISTJE12  S N                               1
       ISTJE12  SEQ4XY             ISTJE12  S Y                           99999
       ISTJE12  SEQ4X1             ISTJE12  S N                           99999
       ISTJE12  SEQ12              ISTJE12  S Y                             500
       ISTJE11  SEQZX              ISTJE11  S N                              33
       ISTJE12  SEQZV              ISTJE12  S N                              33
```

Figure 573. Sequence Objects panel (ADB21Q)

The following primary command is valid on this panel:

GRANT
>> Issues a GRANT command on multiple sequences.
>>
>> **Tip:** The GRANT command operates on each row that is displayed in the table. If you want to omit some of the rows before you issue the command, you can use the minus (-) line command to remove rows from the display. The GRANT command operates only on rows that are listed.

The fields on this panel are:

Sel
>> Input field in which you can enter a line command.

Schema
>> The schema of the sequence.

Name
>> Name of the sequence.

Owner
Owner of the sequence.

T (type)
The sequence type. Possible values are:

    S   User-defined sequence

    I   Identity column

    X   DOCID column for base table containing XML data

    A   Alias

C (cycle)
Specifies whether to wrap values after reaching the maximum value (maxvalue) or minimum value (minvalue). Y indicates Yes and N indicates No.

Start value
Indicates the first value for the sequence.

Option S. Table Spaces

The Table Spaces panel displays the table spaces in the DB2 catalog.

Select option S on the System Catalog panel to display the Table Spaces panel, as shown in the following figure.

The following figure shows the Table Spaces panel.

```
 DB2 Admin ------------------ DB2X Table Spaces -------------------- Row 1 of 5
 Command ===>                                              Scroll ===> PAGE

 Commands: GRANT  MIG  DIS  STA  STO  ALL
 Line commands:
  T - Tables  D - Database  A - Auth  G - Storage group  ICS - Image copy status
  DIS - Display table space  STA - Start table space  STO - Stop table space
  ? - Show all line commands

 Select Name     DB Name   Parts Bpool  L E S I C Tables  Act. pages  Segsz T L
        *        *               * *    * * * * *    *             *   * * *
 ------ -------- -------- ------ ------ - - - - - ------ ----------- ------ - -
 DIS    DSN8S81D DSN8D81A      0 BP0    P N A N N      1          12      0   Y
        DSN8S81E DSN8D81A      4 BP0    P N A N N      1         120      0   Y
        DSN8S81R DSN8D81A      0 BP0    P N A N N      6           0      0   Y
        DSN8S81P DSN8D81A      0 BP0    R N A N N      4          24      4   Y
        DSN8S81S DSN8D81A      0 BP0    P N A N N      1           0      0   Y
 ****************************** END OF DB2 DATA ******************************
```

Figure 574. Table Spaces panel (ADB21S)

The following primary commands are valid on this panel:

GRANT
Issues a GRANT command on multiple table spaces.

MIG
Issues a MIG command on multiple table spaces.

DIS
Issues a DB2 DISPLAY command on multiple table spaces.

STA
Issues a DB2 START command on multiple table spaces.

STO
>Issues a DB2 STOP command on multiple table spaces.

ALL
>Lists all objects of a specified type for each object in a list of objects. Supported values for this panel are:

>**ALL T** Shows all tables for the listed table spaces. Views or aliases are not shown.

>**ALL K**
>>Shows all packages for the listed table spaces.

>**ALL X** Shows all indexes for the listed table spaces.

If the size of the statements generated by the GRANT, DIS, STA, or STO primary command exceeds 32K (an ISPF limit), you will be prompted to send the statements to a batch job or a work statement list (WSL).

If the number of statements generated by the DIS, STA, or STO primary command exceeds 10, you will be prompted to send the statements to a batch job or a WSL.

Restriction: The DROP line command does not allow implicit LOB table spaces to be dropped, but it does allow explicit LOB table spaces to be dropped. This restriction protects you from leaving a definition incomplete.

Recommendation: Primary commands operate on each row that is displayed in the table. If you want to omit some of the rows before you issue the primary command, use the minus (-) line command to remove rows from the display. The primary commands operate only on rows that are listed.

The fields on this panel are:

SELECT
>Input field where you enter one of the line commands listed on the panel.

NAME
>Name of the table space.

DB NAME
>Name of the database.

PARTS
>Number of partitions for a table space. For non-partitioned table spaces, this value is 0.

>To display detailed information for a table space, issue the SP line command against that table space. To display the data set name for the table space (or the data set names for every partition of a partitioned table space), issue the DSN line command against that table space. You can also use the DSN line command against a single partition after you issue the SP line command to display the data set name for that partition only.

BPOOL
>Name of the buffer pool used for the table space.

L Locking size, which is one of the following values:
>**A**   Any
>**L**   Large object (LOB)
>**P**   Page
>**R**   Row

S Table space
T Table
X Implicitly created XML table space

E Erase rule, which is one of the following values:
Y Erase
N No erase

S Status of the table space, which is one of the following values:
A Available
C Incomplete, part index
P Check pending
S Alt check pending
T incomplete, table

I Implicit (whether the table space was created implicitly), which is one of the following values:
Y Yes
N No

C Close rule, which is one of the following values:
Y Yes
N No

TABLES
Number of tables defined in the table space.

ACT. PAGES
Number of active pages in the table space. This field is 0 if the RUNSTATS utility has not been run.

SEGSZ
Number of pages in each segment of a segmented table space. The value is 0 if the table space is not segmented.

T Type of table space, which is one of the following values:
Blank Normal
G The table space was defined with the MAXPARTITIONS option (a partitioned-by-growth table space) with the underlying structure of a universal table space
I Defined with MEMBER CLUSTER and is not greater than 64 GB
K Defined with MEMBER CLUSTER and can be greater than 64 GB
L Defined as LARGE and can be greater than 64 GB
O Defined as an LOB (large object) table space
P Implicit table space created for XML columns
R Range-partitioned universal table space.

L Log changes, which is one of the following values:
Y Yes
N No
X This LOB or XML table space has the NOT LOGGED attribute. Undo and redo logging for the table space is suppressed. Also, the logging attribute for this LOB or XML table space is linked to the logging attribute of the associated base table space and might not be able to be altered independently. If the logging attribute of the base table space is altered to LOGGED, the logging attribute of the LOB or XML table space will also be altered to LOGGED.

Option T. Tables, Views, and Aliases

The Tables, Views, and Aliases panel displays the tables, views, and aliases in the DB2 catalog.

Select option T on the System Catalog panel to display the Tables, Views, and Aliases panel, as shown in the following figure.

On the Tables, Views, and Aliases panel, you can issue many line commands. Enter a question mark (?) on a row to view all valid line commands. These line commands include:
- The N line command lists constraints on tables.
- The GEN line command enables you to reverse engineer DB2 objects from this panel.
- The MIG line command migrates tables and lists of tables.
- The UTL line command generates JCL that can be run against a table.
- The J (Triggers) line command works on views as well as tables.
- The XML line command, when it is issued against a table that has XML columns, shows the XML tables (see "Viewing XML tables" on page 1018).
- The CLONE line command, when it is issued against a base table with a defined clone, displays the clone table (see "Viewing clone tables" on page 1020.)

The following primary commands are valid on this panel:

DET
> Generates a detail report for tables and related objects.
>
> **Note:** The DET primary command is available for the following table types:
> - C: Clone table
> - G: Created global temporary table
> - H: History table
> - P: Implicit table created for XML columns
> - T: Table
> - X: Auxiliary table
>
> .

GRANT
> Issues a GRANT command on multiple tables and views.

MIG
> Issues a MIG command on multiple tables.

ALL
> Lists all objects of a specified type for each object in a list of objects. Supported values for this panel are:
>
> **ALL S** Shows all table spaces for the listed tables.
>
> **ALL K**
> > Shows all packages for the listed tables.
>
> **ALL X** Shows all indexes for the listed tables.
>
> **ALL A**
> > Shows all aliases for the listed tables.

ALL V

> Shows all first-level views for the listed tables. Views on views will not be shown.

ALL VV

> Show all views for the listed tables, including views on views (for DB2 V8 and V9)

Recommendation: Primary commands operate on each row that is displayed in the table. If you want to omit some of the rows before you issue the primary command, use the minus (-) line command to remove rows from the display. The primary commands operate only on rows that are listed.

The fields on this panel are:

`Sel`
> Input field where you enter one of the line commands listed on the panel.

`Name`
> Name of the table, view, or alias.

`Owner`
> Authorization ID of the owner of the table, view, or alias.

T Type of object, which is one of the following values:
  T   Table
  V   View
  A   Alias
  G   Global temporary table
  X   Auxiliary table

> If the base table containing LOB column(s) is dropped and recreated, the explicit auxiliary table is recreated according to its source definition. Changes to the auxiliary table are not reported. Updates to the auxiliary table are ignored if the base table is not recreated.

  M   Materialized table. A materialized table is similar to a view, in that a full SELECT statement is used to create the materialized table query on a table or a view. A materialized table contains physical data behind it and is maintained by the system or by a user. You can use the REFRESH command to refresh the materialized table data. Only a user-maintained materialized table can contain inserts, deletes, and updates. **Restriction:** When a table contains materialized queries, no ALTER commands can be performed on that table.

> You can create a materialized table using the CREM command against a table or a view. You can also create a new materialized table by issuing the CRE command against an existing materialized table.

> You can alter a regular table to make it be a materialized table. Issue the ALM command against a regular table to change it to a materialized table. You can use the DROPM command against a materialized table to drop a materialized query from the materialized table, changing it to a regular table.

  P   Implicit tables created for XML columns.
  C   Clone table.

`DB Name`
> For a table or a view of tables, the name of the database that contains the table space named in TS NAME field. For a view of a view, a global temporary table or for an alias, this field contains DSNDB06.

TS Name
For a table or a view of one table, the name of the table space that contains the table. For a view of a view, this field contains SYSVIEWS. For an alias, this field contains SYSDBAUT.

Cols
Number of columns in the table or view.

Rows
Total number of rows in the table. If the RUNSTATS utility has not been run or if the rows describe a view or an alias, this field contains a value of -1.

Checks
Number of check constraints defined on the table.

C Access control enforced by: R - Row C - Col B - Both ' ' - NA

Issuing table space commands

You can issue table space commands on the Tables, Views, and Aliases panel by appending up to three characters of the table space command to the S command. For example, to display a table space, issue the S.DIS command against the table whose table space you want to display. In addition, the following table space commands can be issued from the Tables, Views, and Aliases panel without prefacing with "S":

- ICS
- IDS
- DISA
- DISC
- DISL
- DISR
- DIST
- DISU
- STA
- STAFO
- STARO
- STARW
- STAUT
- U.C
- U.CC
- U.CI
- U.C2
- U.DG
- U.E
- U.EN
- U.K
- U.KD
- U.KL
- U.M
- U.N
- U.NA

- U.NB
- U.NC
- U.NL
- U.NR
- U.NW
- U.NX
- U.O
- U.OC
- U.OO
- U.OU
- U.P
- U.Q
- U.R
- U.RR
- U.RT
- U.RX
- U.SM
- U.U
- U.V
- U.VC
- U.VG
- U.VI
- U.VL
- U.VP
- U.VR

To execute commands that are longer than three characters, such as STAFO, you must first issue the S.? command on the Tables, Views, and Aliases panel. This command displays a list of all the valid table space commands. From this list, you can select commands to execute.

Note: Table space commands are not allowed on views.

Viewing XML tables

Use the XML line command against a table that has XML columns to display the XML tables. You issue the XML line command on the Tables, Views, and Aliases panel.

```
DB2 Admin ----------- DB2X Tables, Views, and Aliases ----------- ------

 Commands: GRANT       ALL
 Line commands:
  C - Columns  A - Auth  L - List  X - Indexes  S - Table space  D - Database
  V - Views  T - Tables  P - Plans  Y - Synonyms  SEL - Select prototyping
  ? - Show all line commands

 Sel   Name              Schema   T DB Name  TS Name   Cols      Rows Chks C

 ----- ----------------- -------- - -------- -------- ------ ----------- ---- -
       MYCUST            SMITHAJ  T XMLDB3   XMLTS2     5          -1   0
 XML   MYCUSTOMER        SMITHAJ  T XMLDB    XMLTS      5           6   0
       MYCUSTOMER1       SMITHAJ  T XMLDB    XMLTS1     5          -1   0
****************************** END OF DB2 DATA ******************************
```

Figure 575. The Tables, Views, and Aliases panel (ADB21T) – viewing XML tables

You can issue the BASE line command against an XML table to show its
corresponding base table, as shown in the following figure:

```
DB2 Admin ----------- DB2X Tables, Views, and Aliases ----------- ------

 Commands: GRANT       ALL
 Line commands:
  C - Columns  A - Auth  L - List  X - Indexes  S - Table space  D - Database
  V - Views  T - Tables  P - Plans  Y - Synonyms  SEL - Select prototyping
  ? - Show all line commands

 Sel   Name              Schema   T DB Name  TS Name   Cols      Rows Chks C

 ----- ----------------- -------- - -------- -------- ------ ----------- ---- -
 BASE  XMYCUSTOMER       SMITHAJ  P XMLDB    XMYC0000   3           6   0
       XMYCUSTOMER000    SMITHAJ  P XMLDB    XMYC0001   3           0   0
****************************** END OF DB2 DATA ******************************
```

Figure 576. The Tables, Views, and Aliases panel (ADB21T) – viewing XML base

The corresponding base table is shown in the following figure:

```
DB2 Admin ----------- DB2X Tables, Views, and Aliases ----------- ------

 Commands: GRANT       ALL
 Line commands:
  C - Columns  A - Auth  L - List  X - Indexes  S - Table space  D - Database
  V - Views  T - Tables  P - Plans  Y - Synonyms  SEL - Select prototyping
  ? - Show all line commands

 Sel   Name              Schema   T DB Name  TS Name   Cols      Rows Chks C

 ----- ----------------- -------- - -------- -------- ------ ----------- ---- -
       MYCUSTOMER        SMITHAJ  T XMLDB    XMLTS      5           6   0
****************************** END OF DB2 DATA ******************************
```

Figure 577. The Tables, Views, and Aliases panel (ADB21T) – viewing XML base 2

Issue the XMLR line command against a base table that has XML columns to
display information about the XML columns and the related XML base table.

```
DB2 Admin ----------- DB2X Tables, Views, and Aliases ----------- ------

 Commands: GRANT       ALL
 Line commands:
  C - Columns  A - Auth  L - List  X - Indexes  S - Table space  D - Database
  V - Views  T - Tables  P - Plans  Y - Synonyms  SEL - Select prototyping
  ? - Show all line commands

 Sel   Name               Schema    T DB Name  TS Name   Cols        Rows Chks C

 ----- ----------------- -------- - -------- -------- ------ ----------- ---- -
 XMLR  PJTBXML            SMITHAJ  T PJDBXML  PJTSXML      6          -1
       XPJTBXML           SMITHAJ  P PJDBXML  XPJT0000     3          10
 ***************************** END OF DB2 DATA *****************************
```

Figure 578. The Tables, Views, and Aliases panel (ADB21T) – viewing XML column information

The following panel shows the XML column information and the related XML base table.

```
ADB21TXR  ------------------- DB2X XML cols for: JSMITH.PJTBX Row 1 to 1 of 1
Command ===>                                                 Scroll ===> PAGE

Line commands:  T - Table  C - Column

  XML Table: SMITHAJ.PJTBXML
S Owner     Name              Column
  *         *                 *
- -------- ----------------- -------------------
  SMITHAJ  XPJTBXML          INFO
```

Figure 579. The XML cols panel (ADB21TXR) – XML table column information 2

Viewing clone tables

Use the CLONE line command against a table that has a defined clone to display the clone table. You issue the CLONE line command on the Tables, Views, and Aliases panel.

```
DB2 Admin ----------- DB2X Tables, Views, and Aliases ----------- ------

 Commands: GRANT       ALL
 Line commands:
  C - Columns  A - Auth  L - List  X - Indexes  S - Table space  D - Database
  V - Views  T - Tables  P - Plans  Y - Synonyms  SEL - Select prototyping
  ? - Show all line commands

 Sel   Name               Schema    T DB Name  TS Name   Cols        Rows Chks C

 ----- ----------------- -------- - -------- -------- ------ ----------- ---- -
 clone PJCLNBS3           SMITHAJ  T PJMDBCL  PJTSCLN3     2          -1    0
       PJCLNBS4           SMITHAJ  T PJMDBCL  PJTSCLN4     2          -1    0
       PJCLNALIAS         SMITHAJ  C PJMDBCL  PJTSCLN      2          -1    0
 ***************************** END OF DB2 DATA *****************************
```

Figure 580. The Tables, Views, and Aliases panel (ADB21T) – viewing clone tables

Other line commands that support clone tables include:

BASE   Shows the base table for a clone.

DROP   Drop clone tables.

XCHG

Exchange data between base and clone tables.

Note: To see the complete set of line commands for clone tables, enter the "? - Show all line commands" line command on the ADB21T panel.

Option TR. Trusted Contexts

To display trusted contexts choose the TR option on the System Catalog panel.

The trusted contexts panel

Select option 1 on the DB2 Administration Menu to display the System Catalog panel. Select option AO, and then from the Authorization Options panel, select option TR to access the panel for trusted contexts.

The trusted contexts are shown in the following figure.

Note: The only selection criteria allowed for RO and TR options is Name and Column/Operator/Value.

```
ADB2AN in --------------- DB2X Trusted Contexts ----------- Row 1 to 10 of 10
Command ===>                                               Scroll ===> PAGE

Line commands:
  RO - Roles  ID - Authids  ATTR - Attributes  DR - Definer role  DROP - Drop
  I - Interpretation  COM - Comment  DDL - Generate DDL  GEN - Generate SQL


                        D System   Default   O E A A Created
Sel  Name      Definer  T Authid   Role      T N L U Timestamp
     *         *        * *        *         * * * * *
---- -------> -------> - -------- -------> - - - - ----------------------------
     PJCTXROW SMITHAJ    SMITHAJ  PJROLEOW L Y N N 2008-11-12-10.34.32.643009
     PJTCN    PJROLEOW L PJRN               N N N 2008-10-20-14.42.28.663668
     PJTCX    PJROLEOW L MARLINX  PJRX      Y N N 2008-10-20-10.16.29.124017
     PJTCY    PJROLEOW L MARLINY  PJRY      Y N N 2008-10-20-10.22.17.092977
     PJTCZ    PJROLEOW L MARLINZ  PJRZ      Y N N 2008-10-20-10.55.09.611261
     PJTRCXT2 SMITHAJ    MARLINP  PJROLE2   Y N N 2008-09-26-16.54.37.743776
     PJTRCXT7 PJROLEOW L FAKENAME           N N N 2008-10-17-10.28.52.037965
     PJTSTAT1 PJROLEOW L PJTSTAT1           N N N 2008-10-21-16.15.58.731579
     PJTSTAT2 PJROLEOW L PJTSTAT2           N N N 2008-10-21-16.18.36.182001
     PJTSTATJ PJROLEOW L MARTSTJ            Y N N 2008-10-21-16.46.00.787353
***************************** END OF DB2 DATA ******************************
```

Figure 581. Trusted Contexts panel (ADB2AN)

Use the following line commands from this panel to display trusted contexts information:

RO   Displays the default role, if any, and any roles from associated authorization IDs (panel ADB2ARL)

ID   Displays authorization IDs associated with a trusted context (panel ADB2ANID)

ATTR Displays trusted context attributes (panel ADB2ANAT)

DR   Displays the role which defined the trusted context,if any (panel ADB2ARL)

I   Displays interpretation of an object in SYSCONTEXT (panel ADB2ANI)

DROP Use to DROP a trusted context or attribute (panel ADB26DR)

COM Allows you to create a comment for the trusted context (panel ADB26RT)

CRE Use to create a trusted context (panel ADB26CN)

AL   Use to alter a trusted context (panel ADB26CN)

ADDA
         Use to add an attribute to a trusted context (panel ADB26CN)

ADDI Use to add an AuthID to a trusted context (panel ADB26CN)

DDL   Use to generate DDL

GEN   Use to generate SQL from DB2 catalog

Creating or altering a trusted context

To create a trusted context, enter the CRE line command on panel ADB2AN. To alter a trusted context, enter the AL line command on panel ADB2AN. Fill in the required information in the series of panels that appear (shown in the following figure). An example is given for the CRE command.

```
ADB26CN n ----------------DB2X Create Trusted Context ---------------- 05:30
  Command ===> _____

  CREATE TRUSTED CONTEXT
  Name . . . . . . . _____ > (? to look up existing)

  BASED UPON CONNECTION USING SYSTEM AUTHID
  Authid . . . . . . _____ > (primary authid)

  DEFAULT ROLE
  Role . . . . . . . _____ > (role name)

  WITH ROLE AS OBJECT OWNER AND QUALIFIER
  With owner/qual. . ___               (Yes/No)

  ENABLE/DISABLE
  Initial state  . . ___               (Enable/Disable)

  DEFAULT SECURITY LABEL
  Label  . . . . . . _____           (security label name)
                                                           (continued...)

  Press ENTER to continue with attributes or PF3 to cancel
```

Figure 582. Create Trusted Contexts panel (ADB26CN)

```
ADB26CNA  -----------DB2X Create Trusted Context Attribues ----------- 05:30
 Command ===> _____

 CREATE TRUSTED CONTEXT "TEST"


 ATTRIBUTES(

 Choose one::

 ADDRESS . . .  _____   (IP address)

 ENCRYPTION  . ___                                 (None, Low, or High)

 SERVERAUTH  . _____
                                                  (network security zone)
 JOBNAME . . . _____                             (jobname or job prefix*)

 _ Add more attributes
          )

 Press ENTER to continue with IDs or PF3 to restart attribute definition
```

Figure 583. Create Trusted Context Attribues (ADB26CNA)

```
ADB26CNA  -----------DB2X Create Trusted Context Attribues ----------- 05:30
 Command ===> _____

 CREATE TRUSTED CONTEXT "TEST"


 ATTRIBUTES(

 Choose one::

 ADDRESS . . .  _____   (IP address)

 ENCRYPTION  . ___                                 (None, Low, or High)

 SERVERAUTH  . _____
                                                  (network security zone)
 JOBNAME . . . _____                             (jobname or job prefix*)

 _ Add more attributes
          )

 Press ENTER to continue with IDs or PF3 to restart attribute definition
```

Figure 584. Create Trusted Context IDs (ADB26CNU)

Option V. Views

You can use two different methods to display views.

Displaying views using options V

A view might be created that uses multiple tables (for example, a join of two
tables) that are in different databases or different table spaces. The SYSTABLES row
uses one of the database or table space names from one of the tables to put into
the DBNAME and TSNAME fields.

The ADB21T panel uses the SYSTABLES table to populate rows. So you do not
know if the DBNAME or TSNAME is for all the tables that are used by the view
or for just one table of a join. However, on the ADB21VV panel, the TBNAME and

DBNAME fields display '++++++++' if the view has multiple tables in more than one table space or database. '++++++++' also displays if the view references another view or an MQT.

Select option V on the System Catalog panel to display the DB2 Views panel (ADB21VV) which shows data including the number of tables in the view. Use line commands, D, S, and T, to show all DB2 objects that are dependent to the view, including all the dependent views (a view of a view) and tables.

```
ADB21VV in --------------- DB2 Views                        - Row 1 to 18 of 563

Commands: GRANT MIG UTIL ALL
Line commands:
 C - Columns  A - Auth  L - List  S - Table space  D - Database
 T - Tables  Y - Synonyms  SEL - Select prototyping  DDL - Show DDL
 ? - Show all line commands


                                                               Number of
Sel   Name                Schema   C DB Name  TS Name   Cols    Tables
      *                   *        * *        *         *          *
----- ------------------- -------- - -------- -------- ------ ----------
      VDEPT               DSN81010 N DSN8D10A DSN8S10D     4          1
      VHDEPT              DSN81010 N DSN8D10A DSN8S10D     5          1
      VEMP                DSN81010 N DSN8D10A DSN8S10E     5          1
      VPROJ               DSN81010 N DSN8D10A DSN8S10P     8          1
      VACT                DSN81010 N DSN8D10A DSN8S10P     3          1
      VPROJACT            DSN81010 N DSN8D10A ++++++++     5          2
      VEMPPROJACT         DSN81010 N DSN8D10A DSN8S10P     6          1
      VCONA               DSN81010 N DSN8D10P DSN8S10C     5          2
      VOPTVAL             DSN81010 N DSN8D10P DSN8S10C    11          1
      VDSPTXT             DSN81010 N DSN8D10P DSN8S10C     3          1
      VDEPMG1             DSN81010 N ++++++++ ++++++++     7          4
      VEMPDPT1            DSN81010 N DSN8D10A DSN8S10D     7          1
      VASTRDE1            DSN81010 Y DSNDB06   SYSTSTAB   13          1
      VASTRDE2            DSN81010 N DSN8D10A DSN8S10E   13          1
      VPROJRE1            DSN81010 N DSN8D10A DSN8S10P     8          1
      VPSTRDE1            DSN81010 N DSNDB06   SYSTSTAB   12          1
      VPSTRDE2            DSN81010 N DSNDB06   SYSTSTAB   12          1
      VFORPLA             DSN81010 N DSN8D10A DSN8S10P     7          1
Command ===>                                              Scroll ===> PAGE
```

Figure 585. DB2 Views panel (ADB21VV)

Displaying views using option TV

Select option TV on the System Catalog panel to display the Tables, Views, and Aliases panel with a filter showing only views in the catalog.

```
  DB2 Admin ----------- DB2X Tables, Views, and Aliases ----------- ------

 Commands: GRANT      ALL
 Line commands:
  C - Columns  A - Auth  L - List  X - Indexes  S - Table space  D - Database
  V - Views  T - Tables  P - Plans  Y - Synonyms  SEL - Select prototyping
  ? - Show all line commands

 Sel   Name              Schema   T DB Name  TS Name   Cols     Rows Chks C

 ----- ----------------- -------- - -------- -------- ------ ----------- ---- -
       VDEPT             DSN81010 V DSN8D10A DSN8S10D    4         -1   0
       VHDEPT            DSN81010 V DSN8D10A DSN8S10D    5         -1   0
       VEMP              DSN81010 V DSN8D10A DSN8S10E    5         -1   0
       VPROJ             DSN81010 V DSN8D10A DSN8S10P    8         -1   0
       VACT              DSN81010 V DSN8D10A DSN8S10P    3         -1   0
       VPROJACT          DSN81010 V DSN8D10A DSN8S10P    5          1   0
       VEMPPROJACT       DSN81010 V DSN8D10A DSN8S10P    6         -1   0
       VCONA             DSN81010 V DSN8D10P DSN8S10C    5         -1   0
       VOPTVAL           DSN81010 V DSN8D10P DSN8S10C   11         -1   0
       VDSPTXT           DSN81010 V DSN8D10P DSN8S10C    3         -1   0
       VDEPMG1           DSN81010 V DSN8D10A DSN8S10D    7         -1   0
       VEMPDPT1          DSN81010 V DSN8D10A DSN8S10D    7         -1   0
```

Figure 586. The Tables, Views, and Aliases panel (ADB21T) – displaying views

Option X. Indexes

The Indexes panel displays the indexes in the DB2 catalog.

Select option X on the System Catalog panel to display the Indexes panel, as shown in the following figure.

On the Indexes panel, you can issue the UTL line command or UTL primary command to generate JCL for the utilities that can be run against an index.

```
 DB2 Admin -------------------- DB2X Indexes --------------------- Row 1 of 3
  Command ===>                                           Scroll ===> PAGE

  Commands: DIS  STA  STO  ALL
  Line commands:
   T - Tables  D - Database  G - Storage group  P - Plans  C - Columns
   DIS - Display index space  STA - Start index space  STO - Stop index space
   ? - Show all line commands

                         Index                    Table           C C C C
 Select Index Name       Schema   Table Name      Schema    U  Cols G D L M
        *                *        *               *         *  *    * * * *
 ------ ---------------- -------- ---------------- -------- - ------ - - - -
        XDEPT1           DSN8810  DEPT            DSN8810  P    1 N Y N N
        XDEPT2           DSN8810  DEPT            DSN8810  D    1 N Y N N
        XDEPT3           DSN8810  DEPT            DSN8810  D    1 N Y N N
 ***************************** END OF DB2 DATA *****************************
```

Figure 587. Indexes panel (ADB21X)

The following primary commands are valid on this panel:

DIS
    Issues a DB2 DISPLAY command on multiple indexes.

STA
    Issues a DB2 START command on multiple indexes.

STO
> Issues a DB2 STOP command on multiple indexes.

ALL
> Lists all objects of a specified type for each object in a list of objects. Supported values for this panel are:
>
> **ALL T**   Shows all tables associated with the listed indexes.

If the size of the statements generated by the DIS, STA, or STO primary command exceeds 32K (an ISPF limit) or the number of statements generated exceeds 10, you will be prompted to send the statements to a batch job or a work statement list (WSL).

The fields on this panel are:

SELECT
> Input field where you enter one of the line commands listed on the panel.

INDEX NAME
> Name of the index.

INDEX SCHEMA
> The schema of the index

TABLE NAME
> Name of the table on which the index is defined.

TABLE SCHEMA
> The schema of the table.

U   Unique rule, which is one of the following values:
U	Yes
D	No
P	Primary index
C	Unique constraint
R	Unique non-primary parent key
G	Unique ROWID GENERATED BY DEFAULT
N	Unique where NOT NULL
X	Unique column values used to identify or find XML values associated with a specific row.

COLS
> Number of columns in the key.

CG   This field indicates whether CLUSTER was specified when the index was created. This field contains one of the following values:
Y	Yes
N	No

CD   This field indicates whether the table is clustered by the index. This field contains one of the following values:
Y	Yes, which means that more than 95 percent of the rows are in clustering order.
N	No, which means that 95 percent of the rows, or fewer, are in clustering order.

The entry in this field can be changed by using the RUNSTATS utility.

CL   This field indicates whether the data sets are closed when the index is not in use. This field contains one of the following values:
Y	Yes

N   No

CM Index compression
     Y   Active
     N   Not active

XML indexes

XML indexes use the same DB2 catalog support structure as extended indexes (indexes on expressions.)

- Panel ADB21X supports the extended indexes and columns in SYSINDEXES and SYSINDEXPART.
- The KT line command on panel ADB21X displays the information from SYSKEYTARGETS on panel (ADB21Z).
- Line commands are available to display statistics for catalog tables SYSKEYTARGET* and SYSKEYTGT* in the same way as SYSCOL* statistics tables.
- The XC line command on panel ADB21T supports extended indexes.

The following panels support extended indexes and columns in SYSINDEXES and SYSINDEXPART:

ADB21Z - Key Targets
> Lists the key targets that participate in an extended index definition. Display ADB21Z by issuing the line command KT – Key Targets against a table entry on panel ADB21T.

```
ADB21Z in ------------------- DSN9 Key Targets --------------- Row 1 to 2 of 2
Command ===>                                          Scroll ===> PAGE

Line commands:
 T - Table  X - Indexes  I - Interpret  DI - Distribution stats
 PST - Partition stats  RH - Runstats history  KX - Key expression
 UR - Update runstats


                       Index    Key
Sel Index Name         Owner    Seq O Type Name Derived From         Length N
    *                  *         * * *     *                              * *
--- ------------------ -------- ---- - --------- -------------------->  ------ -
    PJMIX2             SMITHJR     1 A VARCHAR   LEFT(CHARCOL3) ASC        10 N
    PJMIX3             SMITHJR     1 A VARCHAR   RIGHT(CHARCOL,2) || C     21 Y
****************************** END OF DB2 DATA ******************************
```

Figure 588. Key targets panel (ADB21Z)

ADB21ZX - Key Targets for Index
> Lists the key targets that participate in an extended index definition for each of the extended indexes of a table. Display ADB21ZX by issuing the line command 'KT – Key Targets" against an index on panel ADB21X.

```
ADB21ZX   -- DSN9 Key Targets for Index SMITHJR.KAVIX2 ------- Row 1 to 2 of 2
Command ===>                                             Scroll ===> PAGE

Line commands:
 X - Index  I - Interpret  DI - Distribution stats  PST - Partition stats
 RH - Runstats history  KX - Key expression  UR -Update runstats

     Key  Col                                                      Distinct
Sel  Seq  Num O Type Name Length N Derived From                      Values
      *    * * *            * * *                                       *
--- ---- ---- - -------- ------ - ------------------------------ -----------
      1    0 A CHAR        3 N SUBSTR(CHARCOL,1,3) ASC                   3
***************************** END OF DB2 DATA *****************************
```

Figure 589. Key targets for index panel (ADB21ZX)

Option XCU. Index Cleanup

Use the Index Cleanup panel to display information about index cleanup activities in the DB2 catalog.

Select option XCU on the System Catalog panel to display the Index Cleanup panel, as shown in the following figure.

```
  ADBP1XCU -------------- DSNB Index Cleanup ------------------ Row 1 to 5 of 5

 Commands: EDIT
 Line commands:
  I - Interpret

            Index    E M      Start    End
Sel Database Space   D W  M  D Time     Time
     *        *       * *  *  * *        *
--- -------- -------- - - -- -- -------- --------
    JRD      ?        D M  1  1 12.01.00 12.30.00
    JRD%     ?        D M  2  2 12.01.00 12.30.00
    JRDTEMP  ?        D M  1  1 12.01.00 12.30.00
    JRDZZZ   NULL     D M  1  1 12.00.01 12.00.06
    JRDZZZ   NULL     D M  ?  ?  ?          ?
```

Figure 590. Index Cleanup panel (ADBP1XCU)

The following primary command is valid on this panel:

Edit
  Enables edit of the index cleanup entries. You can delete, insert, or modify entries without having to use DB2 data manipulation language (DML).

The following line command is valid on this panel:

Interpret
  Provides information about the state object and timestamp information about the object cleanup.

The following fields are displayed on this panel:

Database
  Name of the database that contains the index.

Index Space
  Name of the index space.

ED Enable and Disable. Specifies whether the row enables or disables cleanup for the specified index space.

MW Month and Week. Used to indicate how the value of the DAY is interpreted:
    M   The value of the DAY column is interpreted as a day of the month.
    W   The value of the DAY column is interpreted as a day of the week.

M Indicates the month in which the time window applies. If this column contains NULL, the time window applies to all months.

D Indicates the time window. Indicates the day of the month, if M is specified in the MW column. Indicates day of the week if W is specified in the MW column, or if the MW column is null. When this column represents the day of the week, 1 is for Monday and 7 is Sunday. If this column contains NULL, the time window applies to every day of the month or to every day of the week.

Start Time
The time of the day at which the row starts to apply cleanup. If this column contains a null value, the row applies cleanup at all times on the specified day.

End Time
The time of the day at which the row ends to apply cleanup. If this column contains a null value, the row applies cleanup at all times on the specified day.

Option Y. Synonyms

The Synonyms panel displays the synonyms in the DB2 catalog.

Select option Y on the System Catalog panel to display the Synonyms panel, as shown in the following figure.

```
DB2 Admin ------------------- DB2X Synonyms ------------------- Row 17 of 47
Command ===>                                              Scroll ===> PAGE

 Line commands:
 T - Table  CRE - Create synonym  DROP - Drop synonym  I - Interpretation
 CREAL - Create alias  D - Database  REP - Report   ALT - Redefine synonym

Select  Synonym            Owner    Table/View         Table/View    Created By
                                     Name               Schema
        *                  *         *                  *             *
------- ------------------ -------- ------------------ ------------- --------
        DEMO_UNICODE       DSCGDB2  DEMO_UNICODE       DSN8810       ISTJE
        DEPT               DSCGDB2  DEPT               DSN8810       ISTJE
        EMP                DSCGDB2  EMP                DSN8810       ISTJE
        EMPPROJACT         DSCGDB2  EMPPROJACT         DSN8810       ISTJE
        PROJ               DSCGDB2  PROJ               DSN8810       ISTJE
        PROJACT            DSCGDB2  PROJACT            DSN8810       ISTJE
        TCONA              DSCGDB2  TCONA              DSN8810       ISTJE
        TDSPTXT            DSCGDB2  TDSPTXT            DSN8810       ISTJE
        TOPTVAL            DSCGDB2  TOPTVAL            DSN8810       ISTJE
        VACT               DSCGDB2  VACT               DSN8810       ISTJE
        VASTRDE1           DSCGDB2  VASTRDE1           DSN8810       ISTJE
        VASTRDE2           DSCGDB2  VASTRDE2           DSN8810       ISTJE
        VCONA              DSCGDB2  VCONA              DSN8810       ISTJE
        VDEPMG1            DSCGDB2  VDEPMG1            DSN8810       ISTJE
        VDEPT              DSCGDB2  VDEPT              DSN8810       ISTJE
        VDSPTXT            DSCGDB2  VDSPTXT            DSN8810       ISTJE
        VEMP               DSCGDB2  VEMP               DSN8810       ISTJE
        VEMPDPT1           DSCGDB2  VEMPDPT1           DSN8810       ISTJE
        VEMPLP             DSCGDB2  VEMPLP             DSN8810       ISTJE
        VEMPPROJACT        DSCGDB2  VEMPPROJACT        DSN8810       ISTJE
        VFORPLA            DSCGDB2  VFORPLA            DSN8810       ISTJE
        VHDEPT             DSCGDB2  VHDEPT             DSN8810       ISTJE
        VOPTVAL            DSCGDB2  VOPTVAL            DSN8810       ISTJE
        VPHONE             DSCGDB2  VPHONE             DSN8810       ISTJE
        VPROJ              DSCGDB2  VPROJ              DSN8810       ISTJE
        VPROJACT           DSCGDB2  VPROJACT           DSN8810       ISTJE
        VPROJRE1           DSCGDB2  VPROJRE1           DSN8810       ISTJE
        VPSTRDE1           DSCGDB2  VPSTRDE1           DSN8810       ISTJE
        VPSTRDE2           DSCGDB2  VPSTRDE2           DSN8810       ISTJE
        VSTAFAC1           DSCGDB2  VSTAFAC1           DSN8810       ISTJE
        VSTAFAC2           DSCGDB2  VSTAFAC2           DSN8810       ISTJE
****************************** END OF DB2 DATA ******************************
```

Figure 591. Synonyms panel (ADB21Y)

The fields on this panel are:

Select
   Input field where you enter one of the line commands listed on the panel.

Synonym
   Synonym for the table or view.

Owner
   Authorization ID of the owner of the synonym.

Table/View Name
   Name of the table or view.

Table/View Schema
   The schema of the table or view.

Created By
   Primary authorization ID of the user who created the synonym.

Option AO. Authorization options

You can use the DB2 Admin System Catalog panel to manage authorizations for objects in the DB2 catalog.

About this task

From the DB2 Admin System Catalog panel, you can display information about the authorizations that were granted for the following database objects:

- Collections
- Columns
- Databases
- Data types
- Functions
- Packages
- Plans
- Resources
- Schemas
- Sequences
- Storage groups
- Stored procedures
- System privileges
- Tables
- Table spaces
- User
- User defined
- Views

To display the authorizations granted on a particular type of database object:

Procedure

1. On the DB2 Admin System Catalog panel, type the two-character AO object option in the **Option** field and press Enter. The authorization options are displayed.
2. Type the two-character option that corresponds to the particular type of object in the **Option** field.
3. Optionally, specify a value in either the **Grantor** or **Grantee** fields of the System Catalog panel.

   Recommendation: For optimum performance when using any authorization option (xA), specify a value in either the **Grantor** or **Grantee** fields of the System Catalog panel.
4. Press Enter.

Example

For example, to display authorization information for databases, type DA in the **Option** field, and press Enter. The Database Authorizations panel, as shown in the following figure, is displayed.

```
ADB2AD in -------------- DB2x Database Authorizations ----------------------

 Commands: REVOKE  GRANT
Line commands:                                     C C D D D D D I L R R R S S S
 R - Revoke  GR - Grant  D - Database              R R B B B I R M O E E E T T T
 I - Interpretation  RE - Grantee role             E E A C M S O A A O C P A A O
 RR - Grantor role                                 T T D T A P P G D R O A R T P
                       G Database  Grant           H A S M R I D D E D G V I T S
Sel Grantor  Grantee  T Name      Date             G B     L N B B   B     R
    *         *       * *         *                * * * * * * * * * * * * * * *
--- -------- ------- - -------- ---------------- - - - - - - - - - - - - - - - -
    ADB      ADB     L  ADBDCH   2004-08-28         G G G G G G G G G G G G G G G
    DPGROTH  DPGROTH L  DBEDB1   2004-09-17         G G G G G G G G G G G G G G G
    DPGROTH  DPGROTH L  DBEDB2   2004-09-17         G G G G G G G G G G G G G G G
    DPGROTH  DPGROTH L  DSQDBCTL 2004-06-18         G G G G G G G G G G G G G G G
    DPGROTH  DPGROTH L  DSQDBDEF 2004-06-18         G G G G G G G G G G G G G G G
    DPGROTH  DPGROTH L  DSQ1STBB 2004-06-18         G G G G G G G G G G G G G G G
    DPGROTH  DPGROTH L  RAADB    2004-06-18         G G G G G G G G G G G G G G G
    DPGROTH  DPGROTH L  RDBIDB1  2004-06-18         G G G G G G G G G G G G G G G
    DPGROTH  DPGROTH L  RDBIDB2  2004-06-18         G G G G G G G G G G G G G G G
    DPGROTH  DPGROTH L  RDBIDB3  2004-06-18         G G G G G G G G G G G G G G G
    DSCGDB2  DSCGDB2 L  DSNDB07  2004-05-24       S G G G G G G G G G G G G G G G
    DSCGDB2  DSCGDB2 L  DSNRGFDB 2004-05-24       S G G G G G G G G G G G G G G G
    DSCGDB2  DSCGDB2 L  DSNRLST  2004-05-24       S G G G G G G G G G G G G G G G
    DSCGDB2  DSCGDB2 L  DSN8D81A 2004-05-24       S G G G G G G G G G G G G G G G
    DSCGDB2  DSCGDB2 L  DSN8D81E 2004-05-25       S G G G G G G G G G G G G G G G
    DSCGDB2  DSCGDB2 L  DSN8D81P 2004-05-24       S G G G G G G G G G G G G G G G
    DSCGDB2  DSCGDB2 L  DSN8D81U 2004-05-25       S G G G G G G G G G G G G G G G
    ISTFL2   ISTFL2  L  TFLDB    2004-07-31         G G G G G G G G G G G G G G G
    ISTJE    ISTJE   L  ISTJED   2004-06-22         G G G G G G G G G G G G G G G
    ISTJE    ISTJE   L  MAPD1    2004-10-25         G G G G G G G G G G G G G G G
    ISTJE    ISTJE   L  MAPD2    2004-10-257        G G G G G G G G G G G G G G G
    ISTJE    ISTJE   L  XXXXX    2004-10-04         G G G G G G G G G G G G G G G
    ISTJE    ISTJE   L  YYYYY    2004-10-24         G G G G G G G G G G G G G G G
    DSCGDB2  PUBLIC  L  DSNDB04  2004-05-24       S Y Y
    DSCGDB2  PUBLIC  L  DSN8D81A 2004-05-24       S Y Y Y Y Y Y Y Y Y Y Y Y Y Y
    DSCGDB2  PUBLIC  L  DSN8D81E 2004-05-25       S Y Y Y Y Y Y Y Y Y Y Y Y Y Y
    DSCGDB2  PUBLIC  L  DSN8D81P 2004-05-24       S Y Y Y Y Y Y Y Y Y Y Y Y Y Y
***************************** END OF DB2 DATA ******************************
```

Figure 592. Database Authorizations panel (ADB2AD)

All of the authorization-related panels are structured similarly to the Database
Authorizations panel. Valid primary commands and line commands are listed at
the beginning part of the panel. Next, detailed authorization information about the
type of database object that you selected is displayed. You enter line commands in
the **Sel** field that is located next to the database objects.

From the authorization-related panels, you can grant and revoke authorizations for
a particular object or for all the objects that are displayed.

Refer to the online help for detailed descriptions of the primary commands, line
commands and fields.

Revoking all authorizations from a user

You can revoke all of the directly held or explicitly granted authorizations from a
user.

About this task

To revoke the authorizations from a user:

Procedure

1. On the DB2 Admin System Catalog panel, type the two-character AO object option in the **Option** field and press Enter.

2. Type the two-character UA authorization option in the **Option** field and specify the name of the user or users from whom to revoke authorizations in the **Grantee** field at the bottom of the panel. Press Enter. The User Authorizations Summary panel, as shown in the following figure, is displayed.

```
ADB2AUS n ----------- DB2X User Authorizations Summary -----------------------

Authorities held by VNDSHL1%
Authority includes SYSADM
Commands:  AU AP ALL AE AI
Line commands:  AU - User Only  AP - All PUBLIC  ALL - All Authorizations
                AE - Explicit to User  AI - Implicit to User

Sel Type              Explicit    Implicit     PUBLIC      Total
--- ---------------- ----------- ----------- ----------- -----------
    System                1           0           1           2
    Storage group         0           0           3           3
    Database              0           0          10          10
    Table space           0           0          30          30
    Table                 0           2         735         737
    Column                0           0           0           0
    Plan                  0           0          79          79
    Collection            0           0          15          15
    Package               0           0         235         235
    Function              0           0          54          54
    Buffer pool           0           0           6           6
```

Figure 593. User Authorizations Summary panel (ADB2AUS)

3. Issue the AU or AE command to display the authorizations that are held by the grantees that you specified. AU shows the authorizations that the specified grantees hold directly, and AE shows the authorizations that the specified grantees were granted explicitly. The User Authorizations panel, as shown in the following figure, is displayed.

```
 ADB2AUD n --------------- DB2X User Authorizations ---------------------------

Commands:  REVOKE  GRANT
Line commands:  A - Auth  I - Interpret  R - Revoke  GR - Grant

S  Grantor  Grantee  T Name                           Authority        Date    WGO
   *        *        * *                              *                *       *
-- -------- -------- - -------------------------- ----------------- ------ ---
   R148286  VNDSHL1  Z (SYSTEM)                       SYSADM           030113 YES
   VNDSHL1  VNDSHL1  D SHLIMR1                         DBADM            030929 YES
   VNDSHL1  VNDSHL1  D DBSHL                           DBADM            031003 YES
   VNDSHL2  VNDSHL1  D DBSHL2                          DBADM            031201 NO
   VNDSHL2  VNDSHL1  D DBSHL2                          DBCTRL           031201 NO
   VNDSHL1  VNDSHL1  T VNDSHL1.VDEPT111                ALL              031202 YES
   K351156  VNDSHL1  T K351156.GROUPCONFIG             ALL              030220 NO
   VNDSHL1  VNDSHL1  T VNDSHL1.VDEPT1                  ALL              030115 YES
```

Figure 594. User Authorizations panel (ADB2AUD)

4. Issue the REVOKE primary command to revoke all of the listed system and user authorities from the listed grantees. The Revoke panel, as shown in the following figure, is displayed to remind you of the significant impact that executing the command can have and to have you confirm whether you really want to execute it.

```
ADB2CONF  -- DB2X Revoke ---------------------------- 18:17

This command revokes all system and user authorizations
from the listed grantees. Other privileges from other
users may also be revoked as the result of a CASCADE
revoke. Choose to execute the command or to return.

Select a choice
  1.  Execute the command
  2.  Return




 F1=Help     F2=Split    F3=Exit     F9=Swap    F12=Cancel
```

Figure 595. Revoke panel (ADB2CONF)

5. Select option 1 to execute the REVOKE command. The SQL is generated and executed if the total size of the generated SQL is less than 32K (approximately 60 REVOKE statements). Otherwise, the Statement Execution Prompt panel is displayed, and you can choose to create a batch job with the statements or add the statements to a work statement list (WSL).

Granting a set of authorizations to a user

When managing authorizations, you might want to give all the authorizations that are held by one user, either those held directly or those granted explicitly, to another user or a list of users.

About this task

To grant all the authorizations that are held by one user to another user:

Procedure

1. On the DB2 Admin System Catalog panel, type the two-character AO object option in the **Option** field and press Enter.
2. Type the two-character UA authorization option in the **Option** field and specify the name of the user from whom to copy authorizations in the **Grantee** field at the bottom of the panel. Press Enter. The User Authorizations Summary panel, as shown in the following figure, is displayed.

```
  ADB2AUS n ----------- DB2X User Authorizations Summary -----------------------

 Authorities held by VNDSHL1%
 Authority includes SYSADM
 Commands:  AU AP ALL AE AI
 Line commands:  AU - User Only  AP - All PUBLIC  ALL - All Authorizations
                 AE - Explicit to User  AI - Implicit to User

 Sel Type             Explicit    Implicit    PUBLIC       Total
 --- ---------------- ----------- ----------- ----------- -----------
     System                    1           0           1           2
     Storage group             0           0           3           3
     Database                  0           0          10          10
     Table space               0           0          30          30
     Table                     0           2         735         737
     Column                    0           0           0           0
     Plan                      0           0          79          79
     Collection                0           0          15          15
     Package                   0           0         235         235
     Function                  0           0          54          54
     Buffer pool               0           0           6           6
```

Figure 596. User Authorizations Summary panel (ADB2AUS)

3. Issue the AU or AE command to display the authorizations that are held by the grantee that you specified. AU shows the authorizations that the specified grantee holds directly, and AE shows the authorizations that the specified grantee was granted explicitly. The User Authorizations panel, as shown in the following figure, is displayed.

```
 ADB2AUD n --------------- DB2X User Authorizations ----------------------------

  Commands:  REVOKE  GRANT
  Line commands:  A - Auth  I - Interpret  R - Revoke  GR - Grant

  S  Grantor  Grantee  T Name                            Authority         Date   WGO
     *        *        * *                               *                 *      *
 -- -------- -------- - -------------------------- ----------------- ------ ---
     R148286  VNDSHL1  Z (SYSTEM)                        SYSADM            030113 YES
     VNDSHL1  VNDSHL1  D SHLIMR1                          DBADM             030929 YES
     VNDSHL1  VNDSHL1  D DBSHL                            DBADM             031003 YES
     VNDSHL2  VNDSHL1  D DBSHL2                           DBADM             031201 NO
     VNDSHL2  VNDSHL1  D DBSHL2                           DBCTRL            031201 NO
     VNDSHL1  VNDSHL1  T VNDSHL1.VDEPT111                 ALL               031202 YES
     K351156  VNDSHL1  T K351156.GROUPCONFIG              ALL               030220 NO
     VNDSHL1  VNDSHL1  T VNDSHL1.VDEPT1                   ALL               030115 YES
```

Figure 597. User Authorizations panel (ADB2AUD)

4. Issue the GRANT primary command. The Grant Privileges panel, as shown in the following figure, is displayed.

```
ADB2AUG ----------------- DB2X Grant Privileges -------------------- 18:20
Command ===>

 Specify grantees to use for all the GRANT statements.
 An "S" preceding the listed privilege indicates the privilege exists
 in the list of authorizations shown on the previous panel. Replace "S"
 with null to avoid granting the privilege.

 GRANT

    S SYSADM          SYSCTRL          SYSOPR
      DBADM           DBCTRL           DBMAINT        PACKADM

 TO

 Grantees ===>                                                       >

 With GRANT option ===>      YES - retains option for each GRANT statement
                             NO  - removes option for all GRANT statements
```

Figure 598. Grant Privileges panel (ADB2AUG)

5. Specify the users to whom you would like to grant authorizations in the
   Grantees field. The SQL is generated and executed if the size of the generated
   SQL is less than 32K. Otherwise, the Statement Execution Prompt panel is
   displayed, and you can choose to create a batch job with the statements or add
   the statements to a work statement list (WSL).

Chapter 29. DB2 Admin commands

There are two types of DB2 Admin commands.

The following topics describe the two types of DB2 Admin commands.

Topics:
- "DB2 Admin primary commands"
- "DB2 Admin line commands" on page 1043

DB2 Admin primary commands

Primary commands are issued from the command line on DB2 Admin panels.

The primary commands are shown in the following table. Most primary commands can be entered on all panels. To determine which commands are available for a particular panel and the correct syntax for those commands, access the help for that panel.

Tip: When you enter a DB2 Admin primary command that has the same name as a TSO command, the TSO command is executed first. To bypass the TSO command processor, enter the primary command with a prefix of the greater than symbol (>), which is a TSO escape character.

Table 30. DB2 Admin primary commands

Command	Alias	Description

?		Allows you to navigate directly to an object.
		
		Syntax:
		• For an external command, the first token must be CAT.
		• For a primary command, a single character is used to identify that the specified command is a catalog navigation command. This single character is defined in panel ADB2P2 "Change DB2 Admin Defaults." The default is a question mark (?).
		Example:
		`?xx qualifier.name`
		`?xx  name`
		
		Where:
		– xx is the object type
		– qualifier is the object qualifier
		– name is the object name
		
		Note 1: Object type is optional. If object type is not specified, then specifying qualifier or name results in a syntax error.
		Note 2: Qualifier is optional. If specified, then the object type must be also specified. Any value that is valid in the owner field of the ADB21 panel can be specified. The first period marks the end of the qualifier.
		Note 3: Name is optional. Any value that is valid in the name field of the ADB21 panel can be specified. If a qualifier is specified, it must be terminated with a period, to distinguish the qualifier from the name.

ALL		Lists all objects of a specified type for each object in a list of objects. For example, for a list of indexes on panel ADB21X, the ALL T command will display all tables associated with those indexes.
BIND		Generates BIND commands for multiple application packages or plans. The BIND commands are created in a work statement list. This command is valid only when packages or plans are displayed.
BINDOPT		Displays the Bind Options panel. From the panel, you can choose bind and rebind options that are not in the DB2 catalog records.
BROWSE	B, BR, BRO, BROW	Browse the current ISPF table.
CMM		Displays the Change Management (CM) panel (ADB2C).
COLUMNS		Performs a column lookup when primary, unique, or foreign key constraints are being added.

Table 30. DB2 Admin primary commands (continued)

Command	Alias	Description
DET		Available on the Tables, Views, Aliases panel (ADB21T), and Packages panel (ADB21K), the DET primary command generates a detail report for tables (and related objects) and packages. The report displays the following types of information for tables and their related objects: • Table details • Column information • Index information • Keys information • Aliases information **Restriction:** The DET primary command is available only for the following table types: • C: Clone table • G: Created global temporary table • H: History table • P: Implicit table created for XMl columns • T: Table • X: Auxiliary table . The package details report displays the following information: • Package details • SQL information • Explain information from package owner's plan table
DB2 *db2 command*		Issues a DB2 command. For example: DB2 -DIS THREAD (*). DB2 can be omitted from the command.
DUTIL		Displays the Display or Terminate Utilities panel (ADB2Z2U2). On the panel, you can view a list of utilities that are running and select utilities to stop running.
DIS		Generates a DB2 command to display information for all objects listed. The command is valid only when databases, table spaces, or indexes are displayed.
EDIT		Enables edit of objects listed in a panel.
EXPLAIN		Changes the DB2 CURRENT EXPLAIN MODE.

Table 30. DB2 Admin primary commands (continued)

Command	Alias	Description

FIND*string* [<u>NEXT</u>|PREV] [*fromcolno tocolno*]		Find a string in the rows that are returned in a table display panel. This command allows you to go directly to a particular string without having to scroll. The command starts at the row on which the cursor is positioned and searches all the columns, or the specified columns, for the specified string. If the string is found, the cursor is placed at the row. The default, NEXT, is to search in a forwards direction. To search in a backwards direction, specify PREV. For example:
		
		`FIND MYUSERID`
		`FIND MYUSERID PREV`
		`FIND MYUSERID 2 4`
		`FIND MYUSERID PREV 2 4`
		
		If the string contains special characters, use quotes around the string. You can specify RFIND to repeat the last FIND command.

FREE		Generates FREE commands for multiple application packages or plans. The FREE commands are created in a work statement list. This command is valid only when packages or plans are displayed.
GEN		Generates SQL for the objects from the DB2 catalog.
GRANT		Generates a GRANT statement for all the objects that are listed. This command is valid only when databases, tables, views, aliases, packages, plans, sequences, stored procedures, user-defined functions, user-defined data types, or authorizations are displayed. The GRANT command is useful on authorization panels when copying authorizations from one user to one or more other users, and the command is valid only when the values in the Grantee column are the same.
HASH		Enables fast access to a row by hashing a key value and storing the hash value in a unique index.
ISPF *ispf statement*		Issues one or more ISPF statements. For example: `ISPF SELECT CMD(MYCMD)`. A semicolon (;) should separate ISPF statements.
LIKE		Switches the LIKE operator ON or OFF for search criteria. This command is valid only on the System Catalog panel (ADB21).
MIG		Performs a migration (MIG) on the displayed objects. This command is valid only when databases, table spaces, or tables are displayed.
ORDER		Defines or modifies the ORDER BY clause in the SQL statement that retrieves data for DB2 Admin panels.

Table 30. DB2 Admin primary commands (continued)

Command	Alias	Description
PANEL *panel name*		Displays the panel whose name is specified.
		The purpose of the PANEL command is to allow installations to extend DB2 Admin with their own panels and then use these panels directly with DB2 Admin. The panel must be designed to be invoked this way. That is, the panel should not be designed to be part of a multi-panel dialog and rely on variables being set in the preceding panels. Otherwise, unpredictable results can occur.
PARMS	PARM	Shows or updates current DB2 Admin parameters.
PLANMGMT		Displays the plan management attributes for the packages.

PRINT TABLE ON FILE *ddname* or PRT TABLE ON FILE *ddname*		Prints the current table to the specified file, for example:
		
		`PRT TABLE ON FILE temp1`
		
		If you don't specify a file name, the default file with the ddname PRINT is used. The specified file must be preallocated with a disposition of OLD, for example:
		
		`tso alloc f(temp1) dsn(temp1.list) old`
		
		After the file is allocated, issue the PRT command.

PROMPT (*options*)	PRMT	Changes DB2 Admin prompt options.
QUALIFIER		Displays the qualifier for the packages.
REBIND		Generates REBIND commands for multiple application packages or plans. The REBIND commands are created in a work statement list and contain only the package or plan name. This command is valid only when packages or plans are displayed. When you specify REBIND, the resulting BIND commands contain only the package or plan name. Specify REBIND FULL to have the resulting BIND commands contain both the package or plan name and all of the parameters.
REFRESH	REF	Refreshes the current ISPF table with data from DB2.
REVOKE		Generates REVOKE statements for all of the system authorities, user authorities, and object authorizations that are listed for the specified grantees. When you issue the REVOKE command, you are prompted to confirm that you really want to execute the command because of the significant impact that the command can have.
REP		Generates a batch job that produces a printable report of the objects in the DB2 catalog.

Table 30. DB2 Admin primary commands (continued)

Command	Alias	Description
SAVE		Saves the Detail report to a data set. The Detail report is generated by the DET primary command or DET line command.
SAVE TABLE AS *name* IN LIB *ddname*		Saves the current ISPF table with the specified name in the specified library. If you do not specify a library name, the default library ISPTABL is used. The ddname must be preallocated to a data set before you use this command.
SCHEMA *schema*		Changes the CURRENT SCHEMA. For example, `SCHEMA ISTJE`
SEARCH	SARG	Performs more sophisticated searches of the ISPF tables than the search arguments or the panel allows. When you use the SEARCH command, DB2 Admin displays a panel with all the columns of the ISPF table. On this panel you can specify searches on individual columns by entering a search operator and a search value for the columns. Valid search operator values include: • Equal to: EQ or = • Greater than: GT or > • Greater than or equal to: GE or >= • Less than: LT or < • Less than or equal to: LE or <= • Not equal to: NE or ¬= When you press END (PF3), a subset of the ISPF table with only the data meeting the search criteria is displayed.
SHOW LIBRARY *ddname* ON PANEL *name*		Shows a member list of the specified library on the specified panel. If you do not specify a library name, the default library ISPTABL is used. If you do not specify a panel name, the default panel DB2ADL is used. The ddname must be preallocated to a data set before you use this command.
SHOW TABLE *name* ON PANEL *name*		Shows the specified table. If you do not specify a panel name, the default panel ADB2DF is used.
SPACE		Shows the amount of space (in KB) that is used for the VSAM page set.
SORT *column names*		Sorts on a column in the current ISPF table. You can place the cursor on the column that you want sorted, instead of specifying a column name. If you do not specify a column name, and the cursor is not in a column, DB2 Admin displays a panel in which you can specify your sort criteria.

Table 30. DB2 Admin primary commands (continued)

Command	Alias	Description
SQL *SQL statement*	A plus sign (+)	Issues one or more SQL statements. For example: `SQL SELECT * FROM MYTABLE`. A semicolon (;) should separate SQL statements. If an SQL statement returns rows, the default table display panel shows the rows.
SQLID *id*	AUTH, AUTHID	Shows or changes the current SQLID. For example: `SQLID ISTJE`.
SSID *xxxx*		Switches to another DB2 SSID. For example: `SSID DSN9`.
STA		Generates a DB2 command to start all objects listed. The command is valid only when databases, table spaces, or indexes are displayed.
STO		Generates a DB2 command to stop all objects listed. The command is valid only when databases, table spaces, or indexes are displayed.
STATUS	STAT	Shows the current status of DB2 Admin and execution control statement statistics.
TBLOPTS		Displays the Alter - Table Options panel (ADBP7TOP), allowing you to modify additional table attributes and specify period definitions for the table. Available only from the Alter Table panel (ADB27C).
WSL		Displays the Manage Work Statement Lists panel (ADB2W).
UTIL		Generates utility JCL for the table spaces of all the databases that are listed.
ZOOM		Collapse or expand a section or all sections.

Related concepts:

"Primary commands" on page 129
Primary commands can be issued from the command line on DB2 Admin panels.

DB2 Admin line commands

Line commands are issued from ISPF table display panels and are directed at a particular row or rows of data.

Specify line commands in the line command area, called the **Select** field, in front of each row.

Two types of line commands are available:
* Special line commands
* General line commands

You also can define your own line commands during installation procedure.

Special line commands

Special line commands that are available for a panel are listed in the line command description area.

A question mark (?) line command indicates that there is not enough room to show all of the line commands. Enter ? in the **Select** column to display a list of all of the line commands available for that panel.

Since the objects listed on a panel have varying attributes, not all of the line commands shown on the panel or its extension panel are applicable to each object. An attempt to issue a line command in such a case results in an error message.

Utility line commands, those commands that allow you to move directly to DB2 utility panels, are prefixed with "U."

The following table shows the special line commands.

Table 31. DB2 Admin special line commands

Command	Description
A	Displays information about authorizations for this object.
AC	Shows accelerators.
ACT	Creates a new active version.
ADD	Adds constraints.
	For the ADBDMT Launchpad panel, ADD adds a utility to the panel.
ADDV	For native SQL procedures: ALTER PROCEDURE ADD VERSION
ADDRI	Adds RI-related tables to a list of tables to migrate.
AH	Schema authorization.
AL, ALTER, ALT	Alters an object.
ALIAS	Shows aliases.
ALM	Modifies a table to be a materialized query table.
AN	Analyzes a change.
AT	Shows accelerated tables.
AUX	Displays associated auxiliary table.
AUXR	Displays associated AUX data column.
B	Binds the object.
BASE	Displays associated base table.
BC	Binds the copy of the object.
BIND	For native SQL procedures: BIND DEPLOY command
BLD	Build options.
BR	Browse the object.
C	Shows the columns for this object.
CA	Shows column authorizations (UPDATE or REFERENCES privileges on individual columns of a table or a view).
CAN	Cancels a change or cancels a thread.
CC	Shows columns referenced in constraint.
CDI	Shows column distribution.
CFK	Create a foreign key for the table.

Table 31. DB2 Admin special line commands (continued)

Command	Description
CH	Shows information about the referential integrity defined for child tables or, on the Change Management panels, shows the changes that use the mask, ignore, or version.
CHK	Shows information about table check constraints.
CHR	Shows information about the referential integrity defined for child relations.
CLONE	Displays the clone table.
COM	Adds a comment on the object.
CON	Shows constraints on table.
COUNT	Displays the current number of rows in the table, as measured by the SQL SELECT COUNT(*) function.
CP	Create a prerequisite change.
CRE	Creates an object.
CREA	Creates an auxiliary table.
CREAL	Creates an alias for the object.
CREM	Creates a new materialized query table using a table or a view.
CRESYN	Creates a synonym for the table.
CRET	Creates a table.
CRETAB	Creates a table in a table space.
CRETS	Creates a table space.
CREV	Creates a view.
CREX	Creates an index on the table.
CREY	Creates a synonym for the table.
CS	Creates a change statement.
CX	Create an index for the table.
D	Shows the database for the object. For the System Administration panels, D deletes the row.
DC	Describes the columns.
DDL	Generates DDL for the object from the DB2 catalog.
DEL	Deletes the row in the ADBDMT Launchpad panel. On the Change Management panels, deletes the change, mask, ignore, version, or version scope.
DEP	Shows the dependencies on an object.
DET	Generates a detail report for tables (and related objects) and packages.
DI	Displays distribution statistics.
DIS	Displays information about the status of the object.
DISA	Displays information about the allocated page sets.
DISC	Displays information about SQL claimers.
DISL	Displays information about locks for this object.
DISR	Displays information about restrictions on use for this object.
DIST	Displays information about threads for this object.

Table 31. DB2 Admin special line commands (continued)

Command	Description
DISU	Displays information about correlation or connection IDs for this object.
DK	Deletes the rows for the package.
DP	Shows the dependencies on an object.
DQ	Deletes the rows for the query number.
DRD	Drops the Restrict on Drop attribute for the table.
DROP	Drops the object or constraint.
DROPM	Drops a materialized query from a materialized query table, changing the materialized query table into a table.
DROPSYN	Drops the synonym for the table.
DRPV	Drop version.
DS	Shows the database structure.
DSN	Displays the data set names for the associated table space or index space.
DSP	Shows the database structure, including plans and packages that are dependent on the table spaces, tables, views, aliases, synonyms, and indexes.
E	Normally, E shows related data types. On some panels, E edits the member (which is indicated on the panel).
EA	Edits the job to analyze the change.
ER	Edits the job to run the change (or the job to promote the change).
EN, ENDI	Shows the connections that are either enabled or disabled for the object.
ENV	Displays the environment variables for the selected object.
F	On the BIND and REBIND panels, frees the object. On all other panels, shows related functions.
FC	Shows the From Column.
FK	Shows information about the referential integrity defined for foreign keys.
FR	Shows explain function table rows.
G	Shows the storage groups for the object.
GEN	Generates SQL for the objects from the DB2 catalog.
GR, GRANT	Grants privileges for the object.
GV	Generates a new version file for the version scope.
H	Shows the homonyms for the object.
I	Shows detailed information about the object. For the System Administration panels, it can also mean insert the row.
ICS	Shows the status of image copies for the object.
IG	Shows the ignores for the object.
IH	Inserts an optimizer hint.
IL	Shows the definition (or the ignore lines) for the ignore.
ILOC	Inserts a location.

Table 31. DB2 Admin special line commands (continued)

Command	Description
ILUM	Inserts LU modes.
IMODE	Inserts a mode.
INS	Inserts a row into a table or inserts a change, mask, ignore, or version scope.
IUSER	Inserts an authorization ID for a user.
J	Shows triggers.
JAR	Shows JAVA or JAR detail.
K	Shows the packages for the object.
KT	Shows key targets.
L	Shows the collection for the object.
	For the tables panels, L shows the rows in the table.
	For the System Administration panels, L lists the catalog.
LA	Adds an index to LISTDEF definition.
LAB	Labels the object.
LISTC	Shows the ICF catalog entries.
LKEY	Shows the limit key values for a partitioned table or a partitioned index.
LOC	Shows the location.
LP	Lists the PLAN_TABLE table for the object.
LPA	List all PLAN_TABLE rows for a package.
LST	Shows statistics for LOB table space.
LU	Shows the LU name.
LUM	Shows the LU modes.
M	Shows the DBRMs for the object.
MA	Shows the masks for the object.
MIG	Migrates the table.
ML	Shows the definition (or the mask lines) for the mask.
MODE	Shows the SYSMODESELECT rows for the location.
O	Shows related stored procedures. On the work statement list panels, runs the work statement list online.
OR	Shows the original change.
P	Shows the plans for the object.
PA	Shows information about the referential integrity defined for parent tables.
PAR	Shows information about the referential integrity defined for parent relations.
PARM	Shows the parameter list.
PK	Shows the primary key for this table.
PL	Shows the package lists for the object.
PQ	Shows the prerequisite changes for the change.

Table 31. DB2 Admin special line commands (continued)

Command	Description
PR	For a change, promotes the associated base version file. For a base version, promotes the base version file.
PST	Shows partition statistics.
PT	Changes the protected status of a base version from off to on, or vice versa, so that the base version cannot be deleted easily.
R	Revokes the privilege for the object.
RB	Rebinds the object.
RC	Recovers (or backs out) the completed change.
RE	Shows the recover change for the change. For the Authorization panels, RE shows the grantee role
REG	Regenerates version.
REM	Comments on the object.
REL	List related objects.
REN	Renames a table. For the Rename Index panel, REN renames an index.
REP	Generates a batch job that produces a printable report for the object from the DB2 catalog.
REPV	For native SQL procedures: ALTER PROCEDURE REPLACE VERSION
RESZ	Resizes page sets.
RH	Shows RUNSTATS history.
RI	Shows the referential integrity constraints for the selected table and its related tables
RIT	Shows the referential integrity constraints for the selected table
RIX	RUNSTATS invalidate dynamic SQL cache for index spaces. Issued from the Listdef Utilities panel (ADB25LU).
RO	For the System Catalog panels, displays the object owner role.
RN	Runs the change.
RR	For the Authorization panels, RR shows the grantor role.
RST	Re-registers a change in INITIAL, DEFINED, or ANALYZED status.
RT	Return type.
RTS	Shows real-time statistics for index space.
RX	RUNSTATS invalidate dynamic SQL cache for table spaces. Issued from the Table Space Utilities panel (ADB21US), the Index Utilities panel (ADB21UX), and the Listdef Utilities panel (ADB25LU).

Table 31. DB2 Admin special line commands (continued)

Command	Description
S	Shows the table spaces for the object.
	For the SQL Statements panels, S shows the column in the result.
	For the System Administration panels, S displays or update the table you selected.
	For the ADB2DDF and ADB2ZD2 panels, S selects the location.
	For the ADBDMT Launchpad panel, S starts the tool on that line.
SA	Sorts in ascending order.
SC	Shows the version scopes.
SD	Sorts in descending order.
SEL	Builds the SQL SELECT statement for this object.
SEQ	Identifies column information.
SM	Displays space statistics for database.
SO	Shows the objects that are defined in the version scope.
SP	Shows the table space's parts.
SQ, SQL	Shows the SQL statements.
SR	Shows explain statement table rows.
SRC	Shows the source code for a stored procedure.
ST	Shows the specific table that is associated with the selected column. For changes, shows the statements in the change.
STA	Starts the object.
STAFO	Forces a start of the object.
STARO	Starts the object for a read operation.
STARW	Starts the object for a read/write operation.
STASP	Starts all spaces for read/write.
STAUT	Starts the object so a DB2 utility can access it (no SQL statements can be issued against the object).
STO	Stops the object.
STOQ	Stops the stored procedure and queues requests.
STOR	Stops the stored procedure and rejects requests.
STOSP	Stops all spaces.
SX	Shows all of the indexes on the selected row's table that use the column name in a key.
T	Shows the tables.
TC	Shows the To column.
TERM	Terminates the utility.
U	Updates the row.
U.x	Generates utility job streams by requesting a utility using one of the codes in the following table.
UM	Update XML modifier data for this XML column
UPD	Updates the row in the ADBDMT Launchpad panel.

Table 31. DB2 Admin special line commands (continued)

Command	Description
UR	Updates the information provided by the RUNSTATS utility.
USER	Shows the user names.
USERD	Example of user-defined line command with DB2.
USERI	Example of user-defined line command with ISPF.
USERP	Example of user-defined line command with panel.
USERS	Example of user-defined line command with SQL.
UT, UTL, UTIL	Runs a DB2 utility job against the object.
V	Shows the views on the object.
VB	Shows the objects that are dependent on this view.
VD	Shows the objects on which view is dependent.
VE	Shows the versions.
VOL	Shows the volumes.
VS	Shows how the view was created or, for a version, shows the version scope.
X	Shows the indexes for the object.
XC	Shows indexes, index columns, and key targets on table.
XCHG	Exchanges data between a base table and its associated clone table.
XCU	Shows index cleanup information. **Restriction:** This command is available only on DB2 for z/OS, V11 systems and higher, that operate in New Function Mode (NFM).
XML	Shows the XML tables created for a base table.
XMLR	Shows XML column information and the related XML table.
XP	Shows the parts of the index.
Y	Shows the synonyms for the object.
any installation-defined command	See the links for related reading.

The following table shows the utility line command codes.

Table 32. DB2 Admin utility line command codes

Command	Description	Valid on panel
U.BP	Change batch job parameters	ADB21S, ADB25L, and others
U.C	Copy full	ADB21S, ADB21X, ADB25L
U.CC	Copy concurrent	ADB21S, ADB25L
U.CI	Copy incremental	ADB21S, ADB25L
U.C2	Copy to copy	ADB21S, ADB21X, ADB25L

Table 32. DB2 Admin utility line command codes (continued)

Command	Description	Valid on panel
U.DG	Define GDG for copy data sets	ADB21S, ADB21X, ADB25L
U.E	Mergecopy	ADB21S, ADB25L
U.EN	Mergecopy–newcopy	ADB21S, ADB25L
U.K	Check index	ADB21S, ADB21X, ADB25L
U.KD	Check data	ADB21S
U.L	Load (with input created from U)	ADB21T
U.LX	Load (with input created from UX or UL)	ADB21T
U.M	Modify	ADB21S
U.N	Repair NOCOPYPEND	ADB21S, ADB21X
U.NA	Repair NOCHECKPEND	ADB21S
U.NB	Repair NORCVRPEND	ADB21S
U.NL	Repair LEVELID	ADB21S
U.O	Reorganize	ADB21S, ADB21X, ADB25L
U.OC	Reorganize with inline copy	ADB21S
U.OI	Reorganize index	ADB25L
U.OO	Reorganize online	ADB21S, ADB25L
U.OU	Reorganize unload only	ADB21S, ADB25L
U.P	Report recovery	ADB21S, ADB21X, ADB25L
U.Q	Quiesce	ADB21S, ADB25L
U.R	RUNSTATS for a table space	ADB21S, ADB21X, ADB25L
U.RB	Rebuild index	ADB21X, ADB25L
U.RI	RUNSTATS for an index	ADB25L
U.RIR	RUNSTATS index report	ADB25L
U.RR	RUNSTATS report	ADB21S, ADB21X, ADB25L
U.RT	RUNSTATS for a table (all)	ADB21S, ADB25L

Table 32. DB2 Admin utility line command codes (continued)

Command	Description	Valid on panel
U.RX	RUNSTATS (invalidate dynamic cache)	ADB21S, ADB21X, ADB25L
U.SM	Standard maintenance	ADB25L
U.TU	Specify template usage	ADB21S, ADB21X, ADB25L, and others
U.U	Unload	ADB21S, ADB21T, ADB25L
U.UL	Unload using UNLOAD utility	ADB21T
U.UX	Unload using reorganization unload external	ADB21T
U.V	Recover	ADB21S, ADB21X, ADB25L
U.VC	Recover to copy	ADB21S
U.VG	Recover to last GDG	ADB21S
U.VI	Recover index	ADB21S
U.VL	Recover log only	ADB21S, ADB25L
U.VP	Recover to log point	ADB21S, ADB25L
U.VR	Recover to RBA	ADB21S, ADB25L

General line commands

Three general line commands are available: minus (-), equal (=), and slash (/).

Chapter 30. DB2 Admin data type conversions

DB2 Admin supports different data type conversions.

The following tables show the data type conversions that DB2 Admin supports.

X indicates that DB2 Admin supports the data type conversion.

Table 33. DB2 Admin data type conversions, part 1

Original data type:	New data type:												
	sm. int.	int.	float	dec.	char.	vchar.	long vchar.	graph	var. graph.	long vgr.	date	time	time st.
small integer	X	X	X	X^1	X^5	X^5							
integer	X^1	X	X	X^1	X^5	X^5							
float	X	X	X	X									
decimal	X^1	X^1	X	X^1	X^5	X^5					X	X	X
character	X^5	X^5		X^5	X^1	X^1	X				X^2	X^3	X^4
varchar	X^5	X^5		X^5	X^1	X^1	X				X^2	X^3	X^4
long varchar					X^1	X^1	X				X^2	X^3	X^4
graphic								X	X	X			
vgraphic								X	X	X			
long vgraphic								X	X	X			
date					X^5	X^5					X		X
time					X^5	X^5						X	
time stamp					X^5	X^5					X	X	X

Notes:
1. This conversion checks for truncation and number overflows. Displayed during the ALT process and before job submission.
2. Indicates conversions from character, variable-length character, and long variable-length character to date format. Examples of valid load formats include:
   - dd.mm.yyyy
   - mm/dd/yyyy
   - yyyy-mm-dd
3. Indicates conversions from character, variable-length character, and long variable-length character to time format. Examples of valid load formats include:
   - hh.mm.ss
   - hh:mm AM
   - hh:mm PM
   - hh:mm:ss
4. Indicates conversions from character, variable-length character, and long variable-length character to time stamp format. Examples of valid load formats include:
   - yyyy-mm-dd-hh.mm.ss
   - yyyy-mm-dd-hh.mm.ss.nnnnnn

The following table shows further data type conversions that DB2 Admin supports.

An A or a D indicates that DB2 Admin supports the data type conversion. The object action for A is ALTER, and the object action for D is DROP or DROP-SC.

Table 34. DB2 Admin data type conversions, part 2

Original data type:	sm int	int	float	dec	char	vchar	long vchar	big int	dec float (16)	dec float (34)	binary	var binary
small integer								A	A	A		
integer								A	A	A		
float								D	A	A		
decimal								A^2	A	A		
character											A^1	A^1
varchar											A^1	A^1
long varchar						A						
big integer	D	D	D	A					D	A		
dec float (16)	D	D	D	D				D		A		
dec float (34)	D	D	D	D				D	D			
binary											A	A
var binary											A	A

Notes:
1. The original column must be defined as FOR BIT DATA.
2. Due to a potential issue when converting from DECIMAL(19,0) to BIGINT using DB2 ALTER statement, the product instead will perform a DROP along with data conversion in order to detect the data issue. Consult the *DB2 Version 9.1 for z/OS SQL Reference* ALTER TABLE statement for further details.

Attention:
1. If the truncation action chosen on ADB27CT is "Z" or "T", the action will be DROP.
2. If the conditions in the previous note are not met, the action is a DROP-SC.
3. Changing NULL to NOT NULL requires a DROP operation.

Chapter 31. DB2 Admin with a large number of objects

Enterprise Resource Planning (ERP) applications are increasingly using DB2 for z/OS.

These ERP systems typically have a large number of objects, such as 1 000 databases, 10 000 to 30 000 table spaces, and 20 000 to 100 000 tables that have one or more indexes. Administering such large DB2 systems is a challenge, and when you use certain DB2 Admin functions, you must take into account the large number of objects. In addition, the data sets that are allocated for DB2 Admin and ISPF functions must be large enough to accommodate the large number of objects.

Topics:
- "ISPF work data sets"
- "Output data sets for GEN DDL" on page 1056
- "Other recommendations for a large number of objects" on page 1056

ISPF work data sets

DB2 Admin uses ISPF file tailoring services when generating batch jobs. The ISPF services uses preallocated work data sets when generating the JCL for the batch jobs. However, when you generate JCL for many objects, the preallocated ISPF work data sets might not be large enough.

The ISPF work data sets are either allocated by the TSO logon procedure or dynamically allocated based on ISPF customization parameters. When you generate batch jobs for many objects, you might need to have the allocations changed for the data sets with these ISPF DD names:
- `ISPCTLx`: points to the ISPF temporary data set default name `SPFTEMPx.CNTL`
- `ISPWRKx`: points to theISPF temporary data set default name `SPFTEMPx.WORK`

Where *x* represents an ISPF logical screen name

Example: *x* = value 1-9, A-W

The recommended space allocation for these data sets is SPACE=(CYL,(1,5). This space allocation allows for generating batch jobs with 115,000 lines of JCL, using three extents. If you are experiencing space problems (x37 abends), contact your storage administrator to have the space allocations changed for the DD names listed

Note: For additional information on ISPF temporary data sets, see the "Preallocate ISPF temporary data sets to VIO" topic in Chapter 4 of the ISPF Planning and Customizing documentation.

Example: Fixing a RUN CM JCL failure

If you have a RUN CM ABEND*x*37 failure related to the ISPCTL*x* or ISPWRK*x* DDs, you can resolve it in one of the following ways:
- Online: Use ANALYSE to generate RUN WSL
- Batch: Change the SADBSLIB skeleton member ADB2SPFB by modifying it for the default allocation for ISPWRK1 and ISPWRK2, as follows:

```
//ISPWRK1  DD DSN=&&ISPWRK1,DISP=(NEW,DELETE,DELETE),
//            DCB=(RECFM=FB,LRECL=256,BLKSIZE=256,DSORG=PS),
//            SPACE=(CYL,(5,10)),UNIT=SYSALLDA
//ISPWRK2  DD DSN=&&ISPWRK2,DISP=(NEW,DELETE,DELETE),
//            DCB=(RECFM=FB,LRECL=256,BLKSIZE=256,DSORG=PS),
//            SPACE=(CYL,(5,10)),UNIT=SYSALLDA
```

Output data sets for GEN DDL

When you use the DB2 Admin GEN function to generate DDL for objects in the DB2 catalog, you can choose to place the DDL in different types of output data sets.

When you use the DB2 Admin GEN function to generate DDL for objects in the DB2 catalog, you can choose to place the DDL in:
- An existing or new data set
- An existing or new work statement list (WSL) data set

When you generate DDL for a large number of objects and specify that a new data set be used, either a regular data set or a WSL data set, the default space allocation that DB2 Admin uses might not be sufficient.

If you are experiencing x37 abends on the output data set (either regular or WSL) for the generated DDL, use a preallocated data set instead of a new data set. Define the DDL output data set with the following attributes:
    RECFM=FB
    LRECL=80

The generated DDL for all the objects in an ERP system can get very large, for example, 3 million lines of statements. The GEN DDL output data set for that number of statements would require 287 cylinders. You can use ISPF option 3.2 to preallocate a large data set. A WSL data set must be a partitioned data set.

Other recommendations for a large number of objects

You should follow certain recommendations when you use DB2 Admin in an environment that has a large number of objects.

The following recommendations will help you use DB2 Admin with a large number of objects:
- Reduce the number of objects for primary commands. Running DB2 Admin primary commands on a very large number of objects can take some time and locks your ISPF session while the objects are being processed. If possible, when searching for objects in the DB2 catalog (DB2 Admin option 1), limit the number of objects by specifying a narrower search criteria.
- When searching for objects in the DB2 catalog (DB2 Admin option 1), use a search criteria that allows DB2 to use indexes to retrieve the information that you need. For more information, see the online help for the System Catalog panel (ADB21).
- Add the recommended indexes to the DB2 catalog.
- Run RUNSTATS on the DB2 catalog.
- Ensure that there is free space on the DASD volumes that you are using. DB2 Admin functions might need to expand the data sets beyond the primary allocation. Extending the data sets with secondary extents requires that the DASD volume has sufficient free space. If you are experiencing problems with

space on data sets that have not reached their maximum extents, contact your storage administrator. The storage administrator might need to change the storage policy for these data sets to avoid the problems.

- Ensure that your batch jobs can get sufficient virtual storage. Some DB2 Admin functions keep information in storage while processing through the objects. If you are experiencing out-of-storage abends, specify a large region size on the job card, for example, 64 MB. If you still experience abends, contact your system administrator because the installation limits in the system that you are using might be causing the problem.
- Ensure that your batch jobs can get sufficient CPU time. When you generate the DDL for a large number of objects, you might, depending on your installation settings and processor speed, need to add a TIME=n option on your job card. The recommended initial value for n is 180 (CPU minutes).

Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
J64A/G4
555 Bailey Avenue
San Jose, CA 95141-1003
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

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

programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

IBM, the IBM logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at http://www.ibm.com/legal/copytrade.shtml.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.

Privacy policy considerations

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM's Privacy Policy at http://www.ibm.com/privacy and the section titled "Cookies, Web Beacons, and Other Technologies" in IBM's Online Privacy Statement at http://www.ibm.com/privacy/details. Also, see the "IBM Software Products and Software-as-a-Service Privacy Statement" at http://www.ibm.com/software/info/product-privacy.

Index

Special characters

A

IBM ®