

Version 11 Release 1

*IBM DB2 Object Comparison Tool for
z/OS
User's Guide*

IBM

Version 11 Release 1

*IBM DB2 Object Comparison Tool for
z/OS
User's Guide*



Note:

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

Sixth Edition (August 2015)

This edition applies to Version 11 Release 1 of IBM DB2 Object Comparison Tool for z/OS (product number 5655-DOC) and to all subsequent releases and modifications until otherwise indicated in new editions.

This edition replaces SC19-4139-04.

© **Copyright IBM Corporation 2001, 2015.**

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

Contents

| | | | |
|--|-----------|--|-----------|
| About this information | v | Using Tools Customizer in a multiple-LPAR environment | 54 |
| Chapter 1. DB2 Object Comparison Tool overview | 1 | Optional: Prepare ADBL CLIST | 55 |
| What's new in DB2 Object Comparison Tool | 1 | Optional: Editing the GOCFB2VB job | 55 |
| What does the DB2 Object Comparison Tool do? | 2 | Optional: Customizing JCL | 57 |
| Solutions for managing changes | 2 | Optional: Conforming DB2 Administration Tool data set names | 57 |
| DB2 Object Comparison Tool features and benefits | 3 | Performance considerations | 59 |
| DB2 Object Comparison Tool components and processes | 5 | Optional: Making Object Comparison Tool available from DB2 Administration Tool | 59 |
| DB2 Object Comparison Tool terminology | 6 | Chapter 4. Creating DB2 Object Comparison Tool batch jobs | 61 |
| DB2 Object Comparison Tool scenarios | 9 | DB2 Object Comparison Tool Menu options | 62 |
| Scenario 1: Comparing a development catalog to a production catalog | 9 | The Version File Conversion utility | 63 |
| Scenario 2: Undoing changes | 13 | Specifying source object definitions to be compared | 64 |
| Scenario 3: Comparing DDL to a catalog. | 14 | Specifying that the source is a DDL file | 65 |
| Scenario 4: Copying objects | 16 | Specifying that the source is from an extract from the DB2 catalog | 67 |
| Service updates and support information | 18 | Specifying the input version file data set name | 79 |
| Product documentation and updates | 18 | Specifying that the source is a version scope | 80 |
| Accessibility features | 19 | Exclude objects from the compare process | 82 |
| Chapter 2. Preparing to customize DB2 Object Comparison Tool | 21 | Creating or managing exclude specifications through Change Management | 84 |
| Set up your environment prior to customization | 23 | Creating exclude specifications with an initial compare | 85 |
| Worksheets: Gathering required data set names | 24 | Creating exclude specifications from stored compare results | 86 |
| Worksheets: Gathering parameter values for Tools Customizer | 24 | Specifying target objects to be compared. | 88 |
| Chapter 3. Customizing DB2 Object Comparison Tool | 29 | Specifying the compare masks data set | 90 |
| Starting and preparing Tools Customizer for use | 29 | Specifying compare ignore fields | 95 |
| Best Practice: SMP/E and runtime libraries maintenance strategy for Tools Customizer | 29 | Selecting SYSINDEXPART ignore fields | 98 |
| Starting Tools Customizer | 33 | Selecting SYSDATABASE ignore fields | 99 |
| Modifying Tools Customizer user settings | 34 | Selecting SYSCOLUMNS ignore fields | 99 |
| Hiding and displaying panel text | 38 | Creating ignore changes specifications | 101 |
| Customizing DB2 Object Comparison Tool | 38 | Managing ignore changes specifications | 102 |
| Roadmap: Customizing DB2 Object Comparison Tool for the first time | 39 | Managing ignore specifications through Change Management | 104 |
| Roadmap: Recustomizing DB2 Object Comparison Tool | 40 | Generating a compare batch job | 105 |
| Specifying the metadata library for the product to customize | 41 | Saved compare results | 107 |
| Creating and associating DB2 entries | 42 | Making compare changes through Change Management. | 111 |
| Defining parameters | 44 | Generate compare job options | 113 |
| Generating customization jobs | 48 | Implicit LOB and XML table support | 131 |
| Submitting customization jobs | 48 | Enable authorization switching | 132 |
| Browsing parameters | 50 | Generating a compare batch job for a multi-target import. | 132 |
| Copying DB2 entries | 50 | Specifying masks for a multi-target import | 134 |
| Removing DB2 entries. | 52 | Updating an associated target | 135 |
| Deleting DB2 entries | 52 | Running a work statement list. | 136 |
| Displaying customization jobs | 53 | Saving dialogs | 138 |
| Maintaining customization jobs. | 53 | Managing and restoring dialogs | 139 |
| | | Comparing multiple sources and targets | 140 |

| | | | |
|---|------------|--|------------|
| Chapter 5. Batch DDL file extraction program | 147 | Translation mask names | 192 |
| Supported SQL statements | 147 | Ignore fields. | 194 |
| Batch DDL file extraction program report format | 150 | Generic ignore field specifications | 196 |
| | | XMLMODIFIER ignore field specifications. | 198 |
| | | | |
| Chapter 6. Batch DB2 catalog extraction program | 153 | Chapter 12. DB2 catalog records and associated masks | 201 |
| Batch DB2 catalog extraction program report | 153 | | |
| | | Chapter 13. Specifying alternate input to the generate apply job program .. | 211 |
| Chapter 7. Batch compare program | 157 | Alternate values for the generate apply program | 212 |
| Compare version files | 158 | Creating user-defined templates | 227 |
| Special considerations for comparing DB2 objects | 158 | | |
| Changing or unloading tables with LOBs | 165 | | |
| | | Chapter 14. Comparing a large number of objects | 231 |
| Chapter 8. Creating a Change Management batch job to run compare | 167 | Chapter 15. Reference | 233 |
| Specifying source objects to be compared | 167 | DB2 Object Comparison Tool parameters | 233 |
| Specifying target objects to be compared | 167 | Customization jobs generated by Tools Customizer | 233 |
| Specifying the compare masks. | 167 | Tools Customizer terminology and data sets | 234 |
| Specifying compare ignore fields | 168 | Tools Customizer terminology. | 234 |
| Specifying compare ignore changes | 168 | Data sets that Tools Customizer uses during customization | 236 |
| Specifying compare exclude | 168 | | |
| | | Chapter 16. Troubleshooting and messages | 241 |
| Chapter 9. DB2 Object Comparison Tool condition codes | 169 | Tools Customizer troubleshooting | 241 |
| | | Gathering diagnostic information. | 241 |
| | | Determining the trace data set name | 241 |
| Chapter 10. Batch compare report format | 171 | Tools Customizer messages. | 241 |
| Compare DB2 Objects sample report 1 | 174 | | |
| Compare DB2 Objects sample report 2 | 177 | | |
| Compare DB2 Objects sample report 3 | 179 | Notices | 301 |
| Compare DB2 Objects sample report 4 | 181 | | |
| Compare DB2 Objects sample summary conversion report | 184 | | |
| Possible conversion errors | 184 | Index | 305 |
| | | | |
| Chapter 11. Translation masks and ignore fields | 187 | | |
| Translation masks | 187 | | |

About this information

This information describes how to use IBM® DB2® Object Comparison Tool for z/OS® Version 11 Release 1.

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

- Customize your DB2 Object Comparison Tool environment.
- Compare sets of IBM DB2 objects by using DB2 Object Comparison Tool
- Generate reports and jobs by using DB2 Object Comparison Tool

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 Object Comparison Tool overview

IBM DB2 Object Comparison Tool (also referred to as DB2 Object Comparison Tool) compares two sets of DB2 objects (called *source* and *target* objects), generates a variety of reports showing the differences between the objects, and can generate the jobs that are required to apply changes to the target. DB2 Object Comparison Tool for z/OS runs as an extension to the DB2 Administration Tool for z/OS.

Topics:

- “What's new in DB2 Object Comparison Tool”
- “What does the DB2 Object Comparison Tool do?” on page 2
- “Solutions for managing changes” on page 2
- “DB2 Object Comparison Tool features and benefits” on page 3
- “DB2 Object Comparison Tool components and processes” on page 5
- “DB2 Object Comparison Tool terminology” on page 6
- “DB2 Object Comparison Tool scenarios” on page 9
- “Scenario 1: Comparing a development catalog to a production catalog” on page 9
- “Scenario 2: Undoing changes” on page 13
- “Scenario 3: Comparing DDL to a catalog” on page 14
- “Scenario 4: Copying objects” on page 16
- Service updates and support information
- Product documentation and updates
- Accessibility features

What's new in DB2 Object Comparison Tool

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 11.1, December 2014

The major additions and changes include:

- You can create data set templates that you use to save DB2 Admin Change Management batch parameter variables. See *Creating user-defined templates for creation and use information*.

Version 11.1, April 2014

The major additions and changes include:

- You can replace an existing change during registration. For more information see “Making compare changes through Change Management” on page 111.
- You can use the wildcard character, an asterisk (*), in the RENAME syntax. You can rename multiple databases and table spaces at one time. For more information see “Translation masks” on page 187.

Version 11.1, October 2013

The major additions and changes include:

- DB2 Object Comparison Tool supports multi-target changes. On a central system where a compare is run, you can register the changes to the source objects and import those changes to multiple target environments. You can communicate status and other information about the target changes back to the central system, so that database administrators have a single centralized view of the changes across the various target systems. You can use the multi-target import feature when you use batch and online during compare. For more information see “Generating a compare batch job for a multi-target import” on page 132.

What does the DB2 Object Comparison Tool do?

IBM DB2 Object Comparison Tool for z/OS enables you to identify the structural differences that exist between two or more DB2 catalogs, DDL, or version files. Identifying these differences is key for effective change management. DB2 Object Comparison Tool also generates the jobs that are required to apply the changes to align the catalogs.

Because DB2 Object Comparison Tool simplifies the process of synchronizing source and target objects, you can more easily keep your production database in step with your test and development database.

DB2 Object Comparison Tool also enables you to:

- Build a compare operation by using the walk-through option, which provides an end-to-end framework for specifying job options
- Generate target changes to a work statement list (WSL) for propagation to remote sites
- Report differences by using a variety of reports
- Account for intentional differences and naming discrepancies between two sets of objects

Solutions for managing changes

IBM solutions help IT organizations maximize the return on their investment in DB2 and IMS databases while staying on top of some of today's toughest IT challenges. DB2 Object Comparison Tool provides an easy-to-use way to manage changes.

DB2 Object Comparison Tool simplifies the process of comparing objects and their dependents from different sources and then synchronizing the sources, consistently keeping production databases in step with those for testing and development.

Both tools are designed with an easy-to-use ISPF interface that lets you manage and process the DB2 objects, and organize them for better system throughput.

A DBA who is responsible for managing changes tasks should consider the following questions:

- Do you need a simpler way to manage changes?
- Could your team benefit from working together to build changes by managing information that is stored in a single database?

- Could your team benefit from being able to track the history of changes from a list and organize them (with Change Management) for better system throughput and maximum convenience?
- Would automating routine database administration tasks improve overall productivity?

In addition to DB2 Administration Tool, the following tools are related to DB2 Object Comparison Tool. When used in conjunction with DB2 Object Comparison Tool, these tools help provide a complete solution to your DB2 Administration needs:

- DB2 Administration Tool provides a comprehensive set of functions that help DB2 personnel manage their DB2 environments efficiently and effectively.
- DB2 Automation Tool allows database administrators to focus more on optimizing their databases, automating maintenance tasks, and providing statistical history reports for trend analysis and forecasting.
- DB2 High Performance Unload is a DB2 utility for unloading DB2 tables from either a table space or a backup.
- DB2 Table Editor quickly and easily accesses, updates, and deletes data across multiple DB2 database platforms, including the Informix Dynamic Server 9.1 or later.
- DB2 Utilities Suite delivers full support for the significant enhancements for z/OS, such as universal table spaces, not-logged table spaces, clone tables, new data types (DECFLOAT, BIGINT, and VARBINARY), new rounding modes, and pureXML storage.
- DB2 Administration Toolkit for z/OS the SAP Edition (5697-L30) is comprised of DB2 Administration Tool and DB2 Object Comparison Tool.

DB2 Object Comparison Tool features and benefits

DB2 Object Comparison Tool provides solutions to many different types of database administration problems.

When you rely on DB2 for z/OS as your production database for business-critical information, you typically have a duplicate system for testing and development. But keeping these two systems as mirror images can be challenging because new applications, application modifications, or errors can trigger changes in the attributes of DB2 objects in one system, but possibly not in the other.

DB2 Object Comparison Tool can help you keep your production system synchronized with your testing and development systems. With a rich set of functions, DB2 Object Comparison Tool makes it easy to compare existing objects (and dependent objects) from different sources and subsequently synchronize these sources by staging and propagating changes between the environments.

You can use the masking and ignore fields features of DB2 Object Comparison Tool to address intentional differences and naming conventions. These two features allow you to compare only the real differences that might exist between objects.

Using DB2 Object Comparison Tool, you can perform these common comparisons:

- Objects from a DB2 catalog to objects that reside in another DB2 catalog
- A file with data definition language (DDL) to another file with DDL
- A file with DDL to objects in a DB2 catalog

- A file with DDL to a target from the DB2 catalog with objects that are automatically selected (in this case, target objects are extracted automatically if they are in the source)
- A version file to another version file. When you compare a DDL file or an extract from the DB2 catalog, the definitions are extracted into version files before the actual compare process is performed.

When comparing objects from a DB2 catalog, whether you specify objects at the database level, the table space level, or the table level, all dependent objects, such as views and indexes, are included in the comparison. After you select an object, that object and its dependents are extracted from the catalog or DDL file and placed in a sequential data set, which is referred to as a version file (the source). If you select input from DDL, everything in the DDL file is processed. There is no object selection based on type or name. You can then select the object to which the source should be compared. This object is extracted from its catalog, or DDL, and placed in a separate sequential data set version file (the target).

If you are comparing DDL, and your DDL has a table, then only that table is used.

If you select input from DDL, everything in the DDL file is processed. There is no object selection based on type or name.

The following table shows possible DB2 Object Comparison Tool sources and targets. You can make a comparison between any source and target.

| Source | Target |
|----------------------------|---|
| Objects from a DB2 catalog | Objects from a DB2 catalog |
| DDL file | DDL file |
| Version file | Version file |
| | DB2 catalog with objects automatically selected based on the source specification |

A comparison of the source and target data sets results in the creation of a difference file (changes file). DB2 Object Comparison Tool generates a report of these differences. After reviewing the report, you can direct the DDL generated for target changes to apply jobs. Apply jobs can be stored in a work statement list or a PDS file and then propagated to several remote sites. Changes in a test environment can be automatically migrated to the development or production environment.

DB2 Object Comparison Tool offers several unique and significant features that you can use to improve your DB2 environment. Specifically, DB2 Object Comparison Tool:

Helps you keep your production system a mirror image of your test and development systems.

New applications, application modifications, or mistakes can cause DB2 objects in one system to have different attributes from objects in other systems. DB2 Object Comparison Tool lets you compare objects (and dependent objects) between one DB2 catalog and another. DB2 Object Comparison Tool creates a file of the differences that it finds. DB2 Object Comparison Tool can then generate batch jobs to bring the catalogs back into sync.

Supports the use of compare masks so that you can compare objects with different names

You can use compare masks to convert the names of objects so that you are comparing only for actual differences, not for differences in the names of objects. For example, you can specify that all owner IDs that begin with ABC* in the source be converted to DEF* in the target (the asterisk is a wild card).

Provides an option to specify fields to ignore when objects are compared

You can specify ignore fields handle intentional differences between the objects that you are comparing. You probably do not want to build test objects exactly the same as production objects, so you might specify ignores for size, or buffer pools.

Uses DDL files, DB2 catalog, or version files to get the definition of DB2 objects that are to be compared

DB2 object comparison definitions can come from the following sources:

- A DDL file; that is, a file that contains SQL statements (for example, a SPUFI file). The file should contain data definitions (object definitions).
- An extract from the DB2 catalog; that is, an extract of one or more databases, table spaces, or tables and all the dependent objects
- A previously created version file, which is an internal representation of a set of objects. A version file represents a snapshot at a particular point in time. DB2 Object Comparison Tool creates a version file for each source and target. Version files can be used to perform all comparisons.

Produces a report about the objects compared

The batch compare program produces different reports depending on the reporting options that were in effect or chosen by the user.

Can generate jobs to apply to the target any changes that were found or, alternatively, generate a work statement list of the changes

The generate apply jobs function generates either jobs or a work statement list. The work statement list can be subsequently used to apply changes to the target object. It is not necessary to run the apply jobs until you are satisfied with the changes. Changes also can be imported as a Change in the Change Management Database.

Supports undoing of implemented changes

The undo capability uses a version file to restore application objects to a previous version, if you have made changes and need to revert to the original state of the objects.

DB2 Object Comparison Tool components and processes

This section provides short descriptions of DB2 Object Comparison Tool components and processes.

As shown in Figure 1 on page 6, DB2 Object Comparison Tool consists of the following processes and components:

- An ISPF full screen interface for specifying the objects to compare

The DB2 Object Comparison Tool ISPF full screen interface runs under the DB2 Administration Tool. It uses DB2 Administration Tool functions to display panels and run SQL statements. You invoke DB2 Object Comparison Tool from the DB2 Administration Tool main menu panel.

ISPF help panels are available. Enter HELP or press PF1 to display a help panel.

- A DDL extraction program that reads DDL statements and converts them into a version file
- A DB2 catalog extraction program that reads definitions from the catalog into a version file to support the compare process
- A compare program that compares two version files, produces a report to describe any differences that are found, and generates the information that is needed to apply changes to the target
- A program to register the changes with DB2 Administration Tool where you can then analyze and run the job
- A generate apply jobs function that does one of the following operations:
 - Creates the UNLOAD, DROP, CREATE/ALTER, and LOAD jobs that are necessary to apply the changes to the target
 - Creates a work statement list of the required changes

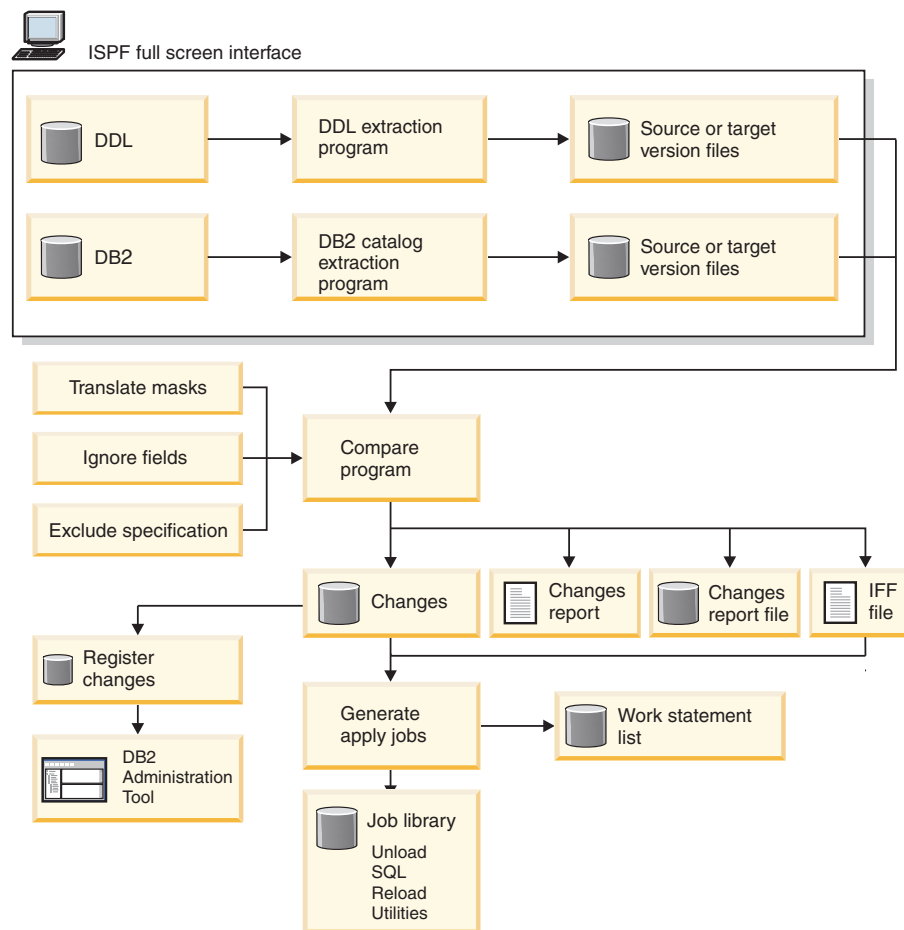


Figure 1. DB2 Object Comparison Tool processes and components

DB2 Object Comparison Tool terminology

DB2 Object Comparison Tool includes several unique terms that you should understand before you begin to use DB2 Object Comparison Tool.

Alternate forms of syntax

Certain functions in DB2 Object Comparison Tool and DB2 Admin support or produce input and output statements that are used by DB2 or by these two products. 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 Object Comparison Tool and 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 supported in any product form.

Changes file

When the source and target data sets are compared, DB2 Object Comparison Tool creates a changes file. DB2 Object Comparison Tool uses the changes file to generate a report of the differences between the source and target. The generate apply jobs function uses the changes file from the compare operation. This file contains:

- The DROP, CREATE, and ALTER statements
- The UNLOAD requests
- Table space information records, which allow the generate apply jobs function to determine the size of the UNLOADs

A typical changes file complete name might be NBRON.PQ76055N.CHANGES.

Exclude

Using Exclude allows you to specify objects to exclude from the inputs of the compare process. An example of an object that can be specified as excluded is a xxtablexxx in a version file. The excluded table does not participate in or affect downstream processing of the version file.

Exclude Specification

Using the exclude specification allows you to list objects to exclude from a compare process. The selected objects are not included as input or output of the compare process.

IFF file

The IFF (Interchange File Format) file is produced by the compare program and is used with the changes file to generate the apply jobs.

Ignore change

On the Ignore Changes panel, you can specify which change to an object is ignored. Ignored changes are reported, but not generated.

Ignore change specification

Use ignore change specifications to select changes to objects to be ignored during compare runs. You select object changes from saved compare results. The selected object types participate in the compare process but changes to the object types are not propagated.

Ignore fields

By using ignore fields, you can compare DB2 catalog records while ignoring some fields. Ignore fields are used in situations where you are aware of differences between source and target objects, but you do not want these changes to be recognized and cause a change.

Source

The structure of the objects as you want them to look. For example, the

source could be the structure of objects in a development environment. A source can be DDL, a version file, or the catalog.

Suppress DROP of objects

By default, DB2 Object Comparison Tool will drop objects from the target that are not in the source. For example, if the source contains only object A, but the target contains both objects A and B, by default DB2 Object Comparison Tool will drop object B.

To change this default behavior, set the Suppress DROP of objects option to Yes. Generally, you should set the Suppress DROP of objects to Yes if your source is a subset of the target and you want to avoid possible dropped objects.

As another example, you specify DDL as the source and you specify a database in the DB2 the catalog as the target. Your catalog contains many tables other than the one table that you are changing. Because all of the additional tables are not in the source, those tables will be dropped unless you specify Suppress DROP of objects =Yes.

Target The destination for the changes. For example, the target could be a production system. The target is where the differences from the source can be applied to make the target the same as the source. The target can be an explicit specification of DDL, a version file, a catalog, or an implicit selection of objects based on the source.

In a situation where you want to change the structure of your production system to match the structure of your development system, the development system is considered the source and the production system is considered the target. In another scenario, you might want to simply identify the differences between two sets of objects, without applying any changes. In this case the source and target represent two different sets of objects that are being compared.

Translation mask

Translation masks are used to allow a match to be found when the compare source and target objects use different naming conventions. Before DB2 Object Comparison Tool compares DB2 catalog record fields, masks are applied to owner and name fields.

Version file

An internal representation of a set of objects. A version file represents a snapshot at a particular point in time. DB2 Object Comparison Tool creates a version file for each source and target. Version files are used to perform all comparisons.

When you compare a DDL file or an extract from the DB2 catalog, the definitions are extracted into version files before the actual compare process is performed.

A version file is created from either a DDL file or from objects in a DB2 catalog. The source version file (with the definition of the objects as they should be) is compared to the target version file (which might be not at the latest level). By using version files as a base, you can compare DDL, catalog object definitions, or other version files in any pairing combination.

A version file is a variable-length data set that contains all the information about the DB2 objects that were extracted. The version file contains a header record and all the DB2 catalog records that represent the objects.

The records in a version file are prefixed with information that allows the compare process to sort the records but at the same time keep multiple records for the same object together.

The advantage of using version files is that they can be saved for subsequent compare operations. For example, you can use them to:

- Restore application objects to a previous version (undo)
- Compare a new version with several production versions (clones) of the objects

Optionally, you can create a version file from the source. This process is useful when the source and the target that you want to compare are not on the same system. In this case you would:

1. Create a version file from the source
2. Transfer the version file to the system where the target is located
3. Specify a compare job where the source is the version file that was transferred and the target comes, for example, from the DB2 catalog

DB2 Object Comparison Tool scenarios

This section provides short descriptions of DB2 Object Comparison Tool common scenarios.

The following scenarios illustrate how DB2 Object Comparison Tool enables you to compare a development catalog to a production catalog, undo changes, compare DDL to a catalog, and copy objects.

Scenario 1: Comparing a development catalog to a production catalog

Comparing a development catalog to a production catalog is one of the most common uses of DB2 Object Comparison Tool.

About this task

Say you have made some changes to your development system. For example, you might have created an extra table or another view, or you might have changed a table or added a column to that table. Or, an application might have changed, and you needed to make a column longer or add a new column. In the following scenario, you are going to compare your development catalog (the source) to your production catalog (the target), and then make the changes in the production catalog to synchronize the two catalogs so that the objects in both catalogs are the same.

The names used in examples are not the names of actual databases.

Tip: Issue the PANELID command so that you can see the name of the panel in the upper left-hand corner.

To compare your development catalog with your production catalog at the catalog level:

Procedure

1. Specify the compare source. In this scenario, your development catalog is the source.

- a. On the GOCMENU main menu panel, select **Option 1 - Specify compare source (new)**.
- b. On the GOC1 Specify Compare Source panel, select **2 - Source is from the DB2 catalog**.
- c. On the GOC12 Specify DB2 Source Catalog Extract panel, select **1 - Source is databases from the DB2 catalog**. Enter the date as part of the **Data set name**, for example devdb.v23.D080311. Enter a description of the source in the **Description** field, for example development database.

Tip: You should have a good naming convention to keep track of the version files. Including the date is one option, there can be others as well.

- d. On the GOC1D Specify Source DB2 Databases panel, insert a line.
 - e. On the GOC1DA Compare Add Databases panel, enter a partial database name. In this scenario, the source databases that you want to compare begin with AGBL, so you enter AGBL, which will display all databases that begin with AGBL.
 - f. Select the specific databases that you want to compare.
2. Specify the compare target. In this scenario, your production catalog is the target.
 - a. On the GOCMENU main menu panel, select **Option 2 - Specify compare target (old)**.
 - b. On the GOC1 Specify Compare Target panel, select **2 - Target is from the DB2 catalog**. Specify the target objects that are going to be included in the target version file.

Tip: You are using **2 - Target is from the DB2 catalog** in this scenario. However, if you select **4 - Target is from the DB2 catalog and objects are automatically selected** you do not have to specify anything. If masking is used, DB2 Object Comparison Tool uses the source object names, in combination with masks and renames that are to be applied, to determine the target object names. DB2 Object Comparison Tool then extracts the target objects accordingly. DB2 Object Comparison Tool will use the source object names, in combination with a mask, if one is used, to determine the target object names, and it will extract the target object names.

- c. On the GOC12 Specify DB2 Target Catalog Extract panel, select **1 - Target is databases from the DB2 catalog**. Enter the date as part of the **Data set name**, for example proddb.v23.D080311. Enter a description of the target in the **Description** field, for example production database. Now you must explicitly specify what target objects are going to be included in the target version file.
 - d. On the GOC1D Specify Target DB2 Databases panel, insert a line.
 - e. On the GOC1DA Compare Add Databases panel, enter a partial database name (for example: DGWD) and a location name (for example: STLEC1).
 - f. Add the databases by selecting them.
3. Specify compare masks.

Often, the names of objects in your development system might not be the same as the names in your production environment. Or the names could be the same, but the owner IDs are different. You could have different DBAs for development and production. You can use compare masks to convert the names of objects when you are doing a compare so that you are comparing only for actual differences, not for differences in the names of objects.

- a. On the GOCMENU main menu panel, select **Option 3, Specify Compare Masks**.
- b. On the Edit Compare Masks panel, specify **Edit Mask** Yes. Note that the **Mask Syntax** is field: inmask, outmask. In **Examples OWNER**, all owners that begin with ABC* in the source will be converted to DEF* in the target (the asterisk is a wild card). So ABCTEST in the source will be converted to DEFTEST in the target. When you specify compare masks, you do not have to specify the same number of characters in both the inmask and in the outmask.
- c. To add masks for your compare, insert a line for each type of mask. For example:

```
OWNER: ABC*,DEF*
DBNAME: *TDB,*PDBA
TSNAME: T*T, P*P
TBNAME: T*,P*
```

Usually, the compare process is iterative. You generate the compare job, then you analyze the report, and look at the differences to see what masks you need to create. If you see messages that indicate that an object was not found or an object was dropped, you might need to provide masks for Owner or Table Names for the next run of the compare job.

Tip: Consider having an experienced DBA set up the masks initially. Other DBAs can then reuse the masks.

4. Generate Compare jobs.
 - a. On the GOCMENU main menu panel, select **5 - Generate compare job**.
 - b. On the GOC5 Generate Compare Jobs panel, use the following settings:

Tip: The settings for parameters are persistent. They remain at the settings that you specified previously.

- Worklist name - TEST
- Suppress DROP of objects - NO
- Suppress DROP of columns - NO
- Suppress adding columns- NO
- Run Validate - Y
- Change reporting options - NO
- PDS for batch jobs - TEST
- Prefix for data sets - TEST
- Generate online - NO
- Generate apply jobs - NO
- Generate one job - YES

Set all the remaining options to NO or N.

- c. On the **JCL** panel, type the submit command (sub) and press Enter.

The job should end with cc=0.

5. Check the report to see the differences between the source and target.
 - a. After the job completes, check the OBJECT COMPARISON REPORT.
 - b. In this report, under **Compare tablespace source and target**, you notice that the STOGROUP name has been changed from AGBLPSG to AGBLTSG. You do not want to change the name of the STOGROUP, so you will need to set a mask for STOGROUP before you run this job again. You also notice that

two other fields were changed, CLOSE and PRIQTY, so you will need to set ignore fields for these two fields. You notice that tables might be altered, dropped, reloaded, and re-created. In the Comparison Counts Report at the bottom of the report, check the stats on the objects that were compared from source and target. Also check the Consistency Checks Report.

Compare tablespace source (AGBLTDB.TBMT001T) and target (AGBLPDB.PBMT001P)
(A)Field CLOSE changed from NO to YES
(A)Field PRIQTY changed from 192 to 48
(A)Field USING changed from 'STOGRUP AGBLPSG' TO 'STOGRUP AGBLTSG'
Tablespace will be altered

Tablespace AGBLPDB.PBMT037P not found on source
Tablespace AGBLPDB.PBMT037P will be dropped

Tablespace AGBLPDB.PBMT0009P not found on target
Tablespace AGBLPDB.PBMT0009P will be added

Compare table source(DBA128.TBMT001_S_M_WORK) and target (DBA128.PBMT001_S_M_WORK)
(A)Add primary key : CD_USER(CD_USER,NO_SEQ)
Tables have identical column lists
Table will be altered

For complete sample reports, see Chapter 10, "Batch compare report format," on page 171.

6. Add the additional masks and ignore fields.
 - a. On the GOCMENU main menu panel, select **Option 3 - Specify Compare Masks**.
 - b. On the Edit Compare Masks panel, add SNAME: *TDB,*PDB.

Tip: Only the first mask that matches a given name is applied, so you should put the most restrictive mask first (*TDB in this example).

- c. On the GOCMENU main menu panel, select **4 - Specify fields to ignore**.
 - d. On the Specify Compare Ignore Fields panel, in the **Data Set Name** field, enter IGNORE5.DATA.
 - e. On the Specify Ignore Fields : Objects panel, select GENERIC, SPACE, and SYSTABLESPACE, and CLOSERULE so that the STOGRUP, CLOSE and PRIQTY fields do not get changed.
 7. Run the job again.
 - a. At this point, DB2 Object Comparison Tool has already created a version file from the objects in the source, so you can use this version file instead of choosing the objects from the catalog. Doing this will save you time and CPU. Version files are compressed to save space.

When you look at the report, you will see the masks and ignore fields listed that you specified in your corrections.

8. Apply the changes to synchronize your systems.
 - a. On the GOC5 Generate Compare Jobs panel, under Options, set **Generate Apply Job = Yes**.
 - b. Run the work statement list to make the actual changes. See "Running a work statement list" on page 136.

What to do next

You have compared a development catalog to a production catalog and you have synchronized your production system with your development system.

Related tasks:

"Scenario 1: Comparing a development catalog to a production catalog" on page 9
Comparing a development catalog to a production catalog is one of the most common uses of DB2 Object Comparison Tool.

“Scenario 2: Undoing changes”

You can use DB2 Object Comparison Tool to undo changes that were made as a result of a catalog to catalog compare.

“Scenario 3: Comparing DDL to a catalog” on page 14

Comparing DDL to a catalog is a common use of DB2 Object Comparison Tool.

“Scenario 4: Copying objects” on page 16

Copying objects is a common use of DB2 Object Comparison Tool.

Scenario 2: Undoing changes

You can use DB2 Object Comparison Tool to undo changes that were made as a result of a catalog to catalog compare.

About this task

You might want to undo the changes that you made in “Scenario 1: Comparing a development catalog to a production catalog” on page 9, and restore the target (the production catalog) to the state that it was in prior to the compare.

To undo changes that were made in a catalog-to-catalog compare:

Procedure

1. Specify the compare source. In this scenario, your source will be the version file that you made of the target.

Tip: Ensure that you save the original version files if you want to be able to undo changes at a later time. Assign descriptive names to your version files so that you can easily retrieve them in the future, and include the date that they were created. For example:

Accounting V9 R10 M08 2008-04-01

- a. On the GOCMENU main menu panel, select **Option 1 - Specify compare source (new)**.
 - b. On the GOC1 Specify Compare Source panel, select **3 - Source is from a compare version file**.
 - c. On the GOC13 Specify Source Compare Version File panel, type in the name of the source version file in the **Data set name** field.
2. Specify the compare target.
 - a. On the GOCMENU main menu panel, select **Option 2 - Specify compare target (old)**.
 - b. On the GOC1 Specify Compare Target panel, select **4 - Target is from the DB2 catalog and objects are automatically selected**.
 3. Do not specify any masks or ignore fields.
 4. Generate compare jobs.
 - a. On the GOCMENU main menu panel, select **5 - Generate compare job**.
 - b. On the GOC5 Generate Compare Jobs panel, use the following settings:
 - Worklist name - TEST
 - Suppress DROP of objects - YES
 - Suppress DROP of columns - NO
 - Suppress adding columns- NO
 - Run Validate - N
 - Change reporting options - NO

- PDS for batch jobs - TEST
- Prefix for data sets - TEST
- Generate online - NO
- Generate apply jobs - NO
- Generate one job - YES

Set all the remaining options to NO or N.

5. Undo the changes that you made previously in the catalog-to-catalog compare, and restore the production catalog.
 - a. On the Generate Compare Jobs panel, under Options, set **Generate Apply Job = Yes**.
 - b. Run the work statement list to restore the production catalog. See “Running a work statement list” on page 136.

What to do next

You have undone the changes that you made in a catalog-to-catalog compare, and restored the production catalog. Note that data added between the time the compare synchronization was done and the time the undo changes process was done might be lost.

Related tasks:

“Scenario 1: Comparing a development catalog to a production catalog” on page 9
Comparing a development catalog to a production catalog is one of the most common uses of DB2 Object Comparison Tool.

“Scenario 2: Undoing changes” on page 13

You can use DB2 Object Comparison Tool to undo changes that were made as a result of a catalog to catalog compare.

“Scenario 3: Comparing DDL to a catalog”

Comparing DDL to a catalog is a common use of DB2 Object Comparison Tool.

“Scenario 4: Copying objects” on page 16

Copying objects is a common use of DB2 Object Comparison Tool.

Scenario 3: Comparing DDL to a catalog

Comparing DDL to a catalog is a common use of DB2 Object Comparison Tool.

About this task

Say you want to generate DDL to change a table on your test system. For example, you might add a column in the middle or at the end of the table. You will generate DDL that shows how the table will look after the change. The DDL contains only the table you want to change. The only thing in the DDL is the CREATE TABLE statement, and it matches an existing table.

Although your table might have an index, a foreign key that is tied to a parent, or other elements, they are not changed because, they are not included in the source DDL.

You will not be adding masks, and the target will be selected automatically. DB2 Object Comparison Tool determines how to change the table. If no table currently exists, DB2 Object Comparison Tool will create the table. If a table does currently exist, DB2 Object Comparison Tool will use the version file and will not go back to the catalog.

DB2 Object Comparison Tool will determine how to change the table. DB2 Object Comparison Tool might have to DROP, re-create, or ALTER a table, depending on the changes that need to be made. DB2 Object Comparison Tool will restore objects and dependencies (such as an index). If the table needs to be dropped and re-created, DB2 Object Comparison Tool will also re-create objects that have been dropped as a result of dropping the table. The table data will be unloaded and, after the object definitions have been applied, reloaded back into the table.

To compare the DDL (the source) and the DB2 catalog (the target):

Procedure

1. Specify the source object definitions to be compared.
 - a. On the GOCMENU main menu panel, select **Option 1 - Specify compare source (new)**.
 - b. On the GOC1 Specify Compare Source panel, select **Option 1 - Source is from a DDL file**.
 - c. On the GOC11 panel, specify the name of the DDL data set.
2. Specify the target objects to which the DDL source will be compared and applied.
 - a. On the GOCMENU main menu panel, select **Option 2 - Specify compare target (old)**.
 - b. On the GOC1 Specify Compare Target panel, select **Option 4 - Target is from the DB2 catalog and objects are automatically selected**.
3. On the GOCMENU main menu panel, select **5 - Generate compare job**. Use the following settings.
 - Worklist name - TEST
 - Suppress DROP of objects - YES
 - Suppress DROP of columns - NO
 - Suppress adding columns - NO
 - Run Validate - N
 - Change reporting options - NO
 - PDS for batch jobs - TEST
 - Prefix for data sets - TEST
 - Generate online - NO
 - Generate apply jobs - NO
 - Generate one job - YESSet all the of the remaining options to NO or N.
4. Run the compare job and check the comparison report.
5. Make any corrections necessary, and generate the compare job again and recheck the comparison report.
6. Apply the changes to finish comparing DDL to a catalog, and change the table in the target.
 - a. On the GOC5 Generate Compare Jobs panel, under Options, set **Generate Apply Job = Yes**.
 - b. Run the work statement list to make the actual changes. See “Running a work statement list” on page 136.

What to do next

You have compared DDL to a catalog, and changed the table in the target.

Related tasks:

“Scenario 1: Comparing a development catalog to a production catalog” on page 9
Comparing a development catalog to a production catalog is one of the most common uses of DB2 Object Comparison Tool.

“Scenario 2: Undoing changes” on page 13
You can use DB2 Object Comparison Tool to undo changes that were made as a result of a catalog to catalog compare.

“Scenario 3: Comparing DDL to a catalog” on page 14
Comparing DDL to a catalog is a common use of DB2 Object Comparison Tool.

“Scenario 4: Copying objects”
Copying objects is a common use of DB2 Object Comparison Tool.

Scenario 4: Copying objects

Copying objects is a common use of DB2 Object Comparison Tool.

About this task

You have a production environment and you want to copy it to make a test environment. Say you have created a new database in your production environment and you want to create images of those objects in the test environment. In this scenario, you will be copying PRODDb to TESTDB, on a different subsystem.

To copy PRODDb to TESTDB, on a different subsystem:

Procedure

1. Specify the source object definitions to be compared.
 - a. On the GOCMENU main menu panel, select **Option 1 Specify compare source (new)**.
 - b. On the GOC12 Specify DB2 Source Catalog Extract panel, select **1 - Source is databases from the DB2 catalog**. Enter the date as part of the **Data set name**, for example proddb.v23.0080311. Enter a description of the source in the **Description** field, for example production database.
 - c. On the GOC1DA Compare Add Databases panel, specify the data set name and the location name (for example STLEC1).
 - d. On the GOC1, select **Option 2 Source is from the DB2 Catalog**, and specify proddb.
2. Specify the target objects.
 - a. On the GOCMENU main menu panel, select **Option 2 Specify compare target (old)**.
 - b. On the GOC1 Specify Compare Target panel, select **4 Target is from the DB2 catalog and the objects are automatically selected**. You choose automatic selection because these objects might already exist.
 - c. On the GOC1DA V81A Compare Add Databases panel, fill in the data set name (for example DGWD) and the location name (for example STLEC1).
3. Specify the masks.

Check to see where names in the test database differ from names in the production database. Owner, table name, or table space names might be

different. The names will vary by environment. You might have to use masking to rename objects. If you are copying a database from a subsystem to another subsystem (not this scenario), you might have to rename all objects.

Tip: When you specify masks, make sure that the first value is the name in the production database and the second value is the name that you want used in the test database. For example: Keyword: *name in Source* (production,*name in Target* (test)).

- a. On the GOCMENU main menu panel, select **Option 3, Specify Compare Masks**.
 - b. On the Edit Compare Masks panel, specify **Edit Mask**Yes.
You might want to overwrite some values, such as COMPRESS. Overwrites are keyed to values.
4. Specify fields to ignore.
- a. On the GOCMENU main menu panel, select **4 - Specify fields to ignore**.
 - b. You probably do not want to build test objects exactly the same as production objects, so you might specify ignores for size, buffer pools, PRIQTY, or SECQTY.
5. Generate compare jobs.
- a. On the GOCMENU main menu panel, select **5 - Generate compare job**.
 - b. On the GOC5 Generate Compare Jobs panel, use the following settings:
 - Worklist name - TEST
 - Suppress DROP of objects - NO
 - Suppress DROP of columns - NO
 - Suppress adding columns- NO
 - Run Validate - N
 - Change reporting options - NO
 - PDS for batch jobs - TEST
 - Prefix for data sets - TEST
 - Generate online - NO
 - Generate apply jobs - NO
 - Generate one job - YESSet all the remaining options to NO or N.
 - c. On the JCL panel, issue the submit command (sub) and press Enter.
The job should end with cc=0.
6. Check the comparison report.
- a. After the job completes, check the **OBJECT COMPARISON REPORT**.
7. Correct any problems with the job by using masks and ignore fields.
8. After you correct the errors, run the apply job to create the new TESTDB.
- a. On the GOC5 Generate Compare Jobs panel, under Options, set **Generate Apply Job = Yes** . When you generate the apply job, if you are modeling a complete set of new objects based on the original objects, data is not loaded or unloaded. No objects are dropped or altered. You are creating the objects, but not populating any data.
 - b. Run the work statement list to make the actual changes. See “Running a work statement list” on page 136.

What to do next

You have copied PRODDb to TESTDB, on a different subsystem.

Related tasks:

“Scenario 1: Comparing a development catalog to a production catalog” on page 9
Comparing a development catalog to a production catalog is one of the most common uses of DB2 Object Comparison Tool.

“Scenario 2: Undoing changes” on page 13
You can use DB2 Object Comparison Tool to undo changes that were made as a result of a catalog to catalog compare.

“Scenario 3: Comparing DDL to a catalog” on page 14
Comparing DDL to a catalog is a common use of DB2 Object Comparison Tool.

“Scenario 4: Copying objects” on page 16
Copying objects is a common use of DB2 Object Comparison Tool.

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 Object Comparison Tool

Before you start to customize DB2 Object Comparison Tool, 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 234 | |
| Software requirements | | |
| Verify that your environment meets the minimum software requirements. To install and use DB2 Object Comparison Tool, your environment must be running a supported version of the z/OS operating system, of DB2 for z/OS, and of DB2 Administration Tool for z/OS. | "Verify that your environment meets software requirements" on page 23 | |
| SMP/E installation | | |
| Verify that DB2 Object Comparison Tool has been installed correctly. DB2 Object Comparison Tool is installed by using standard SMP/E processing. | "Verify that DB2 Object Comparison Tool has been installed successfully" on page 23 | |
| 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 24 | |
| Upgrading to newer versions or modes of DB2 | | |
| No action is required when you migrate from one release or mode of DB2 to another DB2 release or mode. For example, no action is required when you migrate from DB2 V9.1 compatibility mode to DB2 V9.1 new-function mode. No action is required when you upgrade DB2 releases for DB2 Object Comparison Tool because the product relies on DB2 Administration Tool to access DB2. | None. | |
| Gather data set names | | |
| During the customization process, you must specify names for the following types of data sets: <ul style="list-style-type: none"> • Tools Customizer • DB2 Object Comparison Tool | "Worksheets: Gathering required data set names" on page 24 | |
| Gather parameter values | | |

| Task | Link to detailed instructions | Status |
|--|---|--------|
| During the customization process, you must specify parameter values for DB2 Object Comparison Tool, for DB2, and for your LPAR. | "Worksheets: Gathering parameter values for Tools Customizer" on page 24 | |
| Customize DB2 Object Comparison Tool | | |
| Start Tools Customizer by running a REXX EXEC from the ISPF Command Shell panel. | "Starting Tools Customizer" on page 33 | |
| Set up Tools Customizer user settings. If you are running Tools Customizer for the first time, you must modify several user settings to suit your environment. Otherwise, if the user settings that you have already established are still appropriate, skip this step. | "Modifying Tools Customizer user settings" on page 34 | |
| Complete the steps in the appropriate customization roadmap based on the type of customization that you are performing. | | |
| Customizing DB2 Object Comparison Tool for the first time Follow this roadmap if you do not have a customized version of DB2 Object Comparison Tool, and you need to customize it for the first time. | "Roadmap: Customizing DB2 Object Comparison Tool for the first time" on page 39 | |
| Recustomizing DB2 Object Comparison Tool Follow this roadmap if you have a customized version of DB2 Object Comparison Tool but you want to change one or more parameter values. | "Roadmap: Recustomizing DB2 Object Comparison Tool" on page 40 | |
| 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. | | |
| Edit the ADBL CLIST | | |
| Before you can use DB2 Object Comparison Tool, you must allocate the libraries to your ISPF session. | "Optional: Prepare ADBL CLIST" on page 55 | |
| Optional: Edit the GOCFB2VB job | | |
| If necessary, edit this job to specify the correct ADB and GOC SAMP data sets. | "Optional: Editing the GOCFB2VB job" on page 55 | |
| Optional: Customize JCL | | |
| If necessary, customize the DB2 Object Comparison Tool JCL to adhere to your installation standards. | "Optional: Customizing JCL" on page 57 | |
| Optional: Conform DB2 Administration Tool data set names | | |
| Align the DB2 Administration Tool data set names with your local data set naming conventions. | "Optional: Conforming DB2 Administration Tool data set names" on page 57 | |
| Performance considerations | | |
| Understand the factors that can influence DB2 Object Comparison Tool performance. | "Performance considerations" on page 59 | |
| Optional: Making Object Comparison Tool available from DB2 Administration Tool | | |

| Task | Link to detailed instructions | Status |
|--|-------------------------------|--------|
| <p>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 DB2 Object Comparison Tool separately from the customization of DB2 Admin.</p> <p>Follow the customization steps in <i>IBM DB2 Administration Tool for z/OS User's Guide and Reference</i>.</p> | None. | |

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.

Ensure that you are using one of the following supported versions of DB2 for z/OS:

- DB2 V9.1 (5635-DB2) operating in new-function mode
- DB2 Value Unit Edition V9.1 (5697-P12)
- DB2 V10 (5605-DB2)
- DB2 Value Unit Edition V10.1 (5697-P31)
- DB2 V11 (5615-DB2)
- DB2 Value Unit Edition V11.1 (5697-P43)

Ensure that you are using DB2 Administration Tool for z/OS V11.1 (5655-W34).

Verify that DB2 Object Comparison Tool has been installed successfully

See the Program Directory for IBM DB2 Object Comparison Tool for z/OS, GI10-8974 for installation instructions.

To verify the installation was completed correctly, specify the following command on any DB2 Administration Tool panel: `PANEL GOCMENU`. The DB2 Object Comparison Tool menu (GOCMENU) should be displayed. If it does not display, the installation was not successful and you will need to reinstall DB2 Object Comparison Tool.

Important: Because DB2 Object Comparison Tool keeps a large amount of information in memory, it is recommended that a minimum region of 16 MB of memory is used in batch, and 32 MB when running in TSO. In all cases, ensure that the specified region is not being limited via the IEFUSI installation exit. When a large number of objects is processed (more than 1000), additional region is recommended. In addition, more memory is necessary if you suppress object dropping when generating the job because objects are kept resident to process this option. If you are processing extremely large sets of objects (more than 10000), it is recommended that the starting region be at 128 MB. If LE storage failures occur,

increase region parameters before assuming that a problem exists. It is recommended to increase memory in 32 MB increments.

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 | None. | |

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 will 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 236

| 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 customization library. | 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 | | |
| Metadata library The name of the data set that contains the metadata that is used to display the DB2 and LPAR parameters. | DB2TOOL.CCQ110.SCCQDENU | |
| Discover output data set The name of the data set in which the output from the a product or a component Discover EXEC is stored. | DB2TOOL.CCQ110.DISCOVER | |
| Data store data set The name of the data set where Tools Customizer stores information about product or component, LPAR, and DB2 parameter values. | DB2TOOL.CCQ110.DATASTOR | |
| Use DB2 group attach Determines the value that is used in the CONNECT statements in the generated customization jobs. | YES | |
| User Job Card Settings for Customization Jobs | | |
| 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 Object Comparison Tool

Description

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

| Parameter | Sample or default value | Your value |
|---|--------------------------------------|------------|
| Metadata library The name of the DB2 Object Comparison Tool metadata library. | <i>high_level_qualifier.SGOCDENU</i> | |

Product to Customize section

Description

The parameters that are listed in the Product to Customize section are read-only. They contain information that was provided on other panels, by Tools Customizer, or by the DB2 Object Comparison Tool metadata data set.

| | | |
|---|----|---|
| Product metadata library This value is the library that you specified on the Specify the Product to Customize 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 Object Comparison Tool. | No | This value is provided by Tools Customizer. |
| Product name This value displays the product that is being customized. In this example, DB2 Object Comparison Tool 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 <i>Vn.Rn.nn</i> . | 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: Create the VB CLIST and EXEC libraries

Description

If you use CLIST and EXEC libraries that are variable blocked, create VB versions of these libraries. The data set names of the new VB libraries are the same as the FB libraries, but are suffixed with .VB.

This take is optional.

Jobs generated

GOCFB2VB. This job is based on the GOCFB2VB template and is in the *job_sequence_numberFB2VDB2_entry_ID* member.

Required authority

The user ID that runs the GOCFB2VB job must have SYSADM or equivalent authority.

| Step or parameter | Required? | Default value | Your value |
|--|-----------|---------------|------------|
| Create the VB libraries Copy the fixed-block libraries to variable-blocked libraries. | Yes | Selected | |
| DB2 Object Comparison high-level qualifier The high-level qualifier of the DB2 Object Comparison Tool data sets for the product customization step. | Yes | GOCB10 | |
| Fixed to variable blocked VOLSER The volume_serial used for copying fixed-block libraries to variable-blocked libraries. This is not necessary for SMS (System Managed Storage). | Yes | No default | |
| Fixed to variable blocked UNIT The UNIT name used for copying fixed-block libraries to variable-blocked libraries. This is not necessary for SMS (System Managed Storage). | Yes | STD | |
| DB2 Admin customization high-level qualifier The high-level qualifier of the DB2 Admin Tool data sets, one of which contains the EXEC (ADBFVB) which performs the conversion. | Yes | ADBB10 | |

Task: Sample JCL step for CM Batch

Description

Create the sample JCL step for Change Management Batch interface.

This task is optional.

Jobs generated

GOCCM2. This job is based on the GOCCM2 template and is in the *job_sequence_number*CM2 member.

Required authority

None.

| Step or parameter | Required? | Default value | Your value |
|---|-----------|---------------|------------|
| Create the CM Batch interface job step Insert the generated job step in your CM Batch job JCL. | Yes | Selected | |
| DB2 Object Comparison high-level qualifier The volume serial number (VOLSER) used for this task. Specifying a value is not required if managed by Storage Management Subsystem (SMS). | Yes | GOCB10 | |

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).

| Parameter | Required? | 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 | No default | |
| DB2 subsystem ID description A description for the DB subsystem. The value must be 72 characters or less. | Yes | No default | |
| Group attach name The name of the group attach name. | Yes | No default | |

| General DB2 Information | | | |
|---|-----------|---------------|------------|
| Parameter | Required? | Default value | Your value |
| Mode The mode in which the DB2 subsystem is running. The following values are valid: <ul style="list-style-type: none"> • 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 | NFM | |
| Level number The version, release, and modification level of the DB2 subsystem. The following values are valid: <ul style="list-style-type: none"> • 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 | blank | |

Chapter 3. Customizing DB2 Object Comparison Tool

When you customize the DB2 Administration Tool, you can also enable the DB2 Object Comparison Tool for immediate use. At that time, if you choose not to enable the DB2 Object Comparison Tool, you can later customize the tool separately.

Many of the customization tasks are optional.

The following topics provide additional information:

- “Optional: Prepare ADBL CLIST” on page 55
- “Optional: Editing the GOCFB2VB job” on page 55
- “Optional: Customizing JCL” on page 57
- “Optional: Conforming DB2 Administration Tool data set names” on page 57
- “Optional: Making Object Comparison Tool available from DB2 Administration Tool” on page 59

Prerequisite: Before you can customize the DB2 Object Comparison Tool, you must have installed the product by using the instructions in the Program Directory.

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 Object Comparison Tool.

Best Practice: SMP/E and runtime libraries maintenance strategy for Tools Customizer

Tools Customizer creates relationships between the values for the Product Parameters, LPAR Parameters, and DB2 Subsystem Parameters for each Tools Customizer enabled product. Determining the correct maintenance strategy for your Tools Customizer runtime libraries, after SMP/E processing, can reduce problems working with Tools Customizer and the enabled products through their life cycles.

Tools Customizer has very specific requirements for data set names:

- Only one DATASTOR data set exists per LPAR
- The product metadata library data set names do not change during the life of that release of the Tools Customizer enabled product.

The DATASTOR data set is the repository for all the information that Tools Customizer requires to generate customization JCL for enabled products.

When you update and save the Tools Customizer Settings panel (CCQPSET), as described in “Modifying Tools Customizer user settings” on page 34, the name of the DATASTOR data set is saved in the ISPF profile. This allows Tools Customizer to know the active DATASTOR data set when the TSO user id logs in and starts the Tools Customizer EXEC.

Maintenance scenarios

IBM expects maintenance to be applied to libraries which are then used by Tools Customizer. In practice, different customer shops distribute SMP/E APPLY maintenance in different ways.

The following scenarios explains some considerations and alternatives for determining your maintenance strategy. The one overriding objective is to preserve and maintain the same data set names for the Tools Customizer instance.

Apply SMP/E maintenance to the same data sets (using the SMP/E APPLY command):

If you apply SMP/E maintenance (using the SMP/E APPLY command) using the same data set names with each maintenance cycle, you can either use these target libraries as your Tools Customizer runtime libraries or you can copy the SMP/E target data sets to the runtime libraries that are used by Tools Customizer to customize enabled products.

Tools Customizer assumes that if the product metadata library (*DENU) has the same name, this metadata library is for the same release of the enabled product. For example, assume that you customize DB2 Log Analysis Tool v3.3 and name the metadata library SYS2.DB2T00L.SALADENU, with no indication of the version or release. You then upgrade to DB2 Log Analysis Tool v3.4 and employ the same naming convention, SYS2.DB2T00L.SALADENU. Tools Customizer will assume that you are continuing to work with DB2 Log Analysis Tool v3.3 and will report v3.3 on panels and continue to use the same v3.3 Customization Library data set.

A more sustainable approach to naming the data sets is to include a product version, release identifier, or other distinguishing qualifier in the name of the metadata library, so that Tools Customizer can determine the new product release when you upgrade. For example, using metadata library names that include a product version, release identifier, or other distinguishing qualifier, similar to the following, can make product maintenance and upgrades easier:

- DB2T00L.R330.SALADENU for DB2 Log Analysis Tool v3.3
- DB2T00L.R340.SALADENU for DB2 Log Analysis Tool v3.4

Apply SMP/E maintenance to new data sets (using the SMP/E APPLY command):

If you apply SMP/E maintenance (using the SMP/E APPLY command) to new data sets rather than to the same data sets, the next time you open the product metadata library, Tools Customizer will return a data set error that indicates that the library name is being used by another product or component.

For example, assume that you name the DB2 High Performance Unload for z/OS target metadata libraries to reflect the date of an upgrade or to reflect a specific RSU, as follows:

- To reflect a specific upgrade date (August 2014):
 - DB2T00L.PTF420.SINZDBRM.D201408
 - DB2T00L.PTF420.SINZDENU.D201408
 - DB2T00L.PTF420.SINZLOAD.D201408
- To reflect a specific RSU (RSU 1406):
 - DB2T00L.PTF420.SINZDBRM.RSU1406

- DB2TOOL.PTF420.SINZDENU.RSU1406
- DB2TOOL.PTF420.SINZLOAD.RSU1406

Using either of these naming conventions, the next time you start the Tools Customizer EXEC, it will return a data set error.

To handle this type of SMP/E maintenance processing, you can do either of the following:

- Define aliases (using ALIAS control statements) to reference the appropriate libraries for Tools Customizer processing.
- Copy the SMP/E libraries to a set of runtime libraries that are specifically for Tools Customizer processing.

Define aliases (using ALIAS control statements) to reference the appropriate libraries for Tools Customizer processing:

Defining aliases (using ALIAS control statements) for the SMP/E created new product data set names is likely the best strategy when planning for Tools Customizer.

If you are setting up Tools Customizer for the first time, consider specifying the product library data set names with an indicator that these data sets will be used for Tools Customizer processing. For example, use data set names similar to the following names:

- TCZ.PTF420.SINZDBRM
- TCZ.PTF420.SINZDENU
- TCZ.PTF420.SINZLOAD

After applying maintenance using SMP/E, which creates new product library data sets, you should define aliases (using ALIAS control statements) for the new data set names to the data set names that Tools Customizer originally processed. For example, the following maintenance data sets have aliases defined to the original data sets:

- DB2TOOL.PTF420.SINZDBRM.RSU1406 --> TCZ.PTF420.SINZDBRM
- DB2TOOL.PTF420.SINZDENU.RSU1406 --> TCZ.PTF420.SINZDENU
- DB2TOOL.PTF420.SINZLOAD.RSU1406 --> TCZ.PTF420.SINZLOAD

You will need to define an alias (using ALIAS control statements) to each of the following IBM-distributed Tools Customizer data sets:

- SCCQDENU
- SCCQEXEC
- SCCQLOAD
- SCCQMENU
- SCCQPENU
- SCCQSAMP
- SCCQTENU

If you have already set up Tools Customizer and customized the product, you will have to define aliases (using ALIAS control statements) for the newly created data set names to the data sets that were specified when the product was originally customized using Tools Customizer.

After defining the aliases, you should be able to run Tools Customizer successfully.

Note: Only define aliases for IBM-distributed SMP/E libraries.

Restriction: Do not define an alias for any Tools Customizer created data sets, like the following three data sets on the Tools Customizer Settings panel (CCQPSET):

- Customization library qualifier
- Discover output data set
- Data store data set

Do not define an alias for any data sets that you create on behalf of a product, for example the DB2 High Performance Unload for z/OS parameter library (INFPLIB).

Copy the SMP/E libraries to a set of runtime libraries that are specifically for Tools Customizer processing:

If you are setting up Tools Customizer for the first time, consider specifying the product library data set names with an indicator that these data sets will be used for Tools Customizer processing. For example, use data set names similar to the following names:

- TCZ.R420.SINZDBRM
- TCZ.R420.SINZDENU
- TCZ.R420.SINZLOAD

After applying maintenance using SMP/E, which creates new product library data sets, you should copy the new data sets to the data sets that Tools Customizer originally processed. For example, copy the following maintenance data sets to the original data sets:

- DB2TOOL.PTF420.SINZDBRM.RSU1406 --> TCZ.R420.SINZDBRM
- DB2TOOL.PTF420.SINZDENU.RSU1406 --> TCZ.R420.SINZDENU
- DB2TOOL.PTF420.SINZLOAD.RSU1406 --> TCZ.R420.SINZLOAD

You will need to copy each of the following IBM-distributed Tools Customizer data sets:

- SCCQDENU
- SCCQEXEC
- SCCQLOAD
- SCCQMENU
- SCCQPENU
- SCCQSAMP
- SCCQTENU

If you have already set up Tools Customizer and customized the product, you will have to copy the newly created data sets to the data sets that were specified when the product was originally customized using Tools Customizer).

After copying the SMP/E data sets to the Tools Customizer instance libraries, you should be able to run Tools Customizer successfully.

Note: Only copy IBM-distributed SMP/E libraries.

Restriction: Do not copy any Tools Customizer created data sets, like the following three data sets on the Tools Customizer Settings panel (CCQPSET):

- Customization library qualifier
- Discover output data set
- Data store data set

Do not copy any data sets that you create on behalf of a product, for example the DB2 High Performance Unload for z/OS parameter library (INFPLIB).

Related tasks:

Modifying Tools Customizer user settings

Before you can customize a product or a component with Tools Customizer, you must review the settings that Tools Customizer uses.

Related information:

The SMP/E APPLY command

The APPLY command specifies which of the received SYSMODs are to be selected for installation in the target libraries.

Alias processing: SMP/E for z/OS Commands

When an element with aliases is processed, both the element and its aliases are updated. SMP/E does not check the aliases against elements maintained in the target zone.

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 Object Comparison Tool.

Modifying Tools Customizer user settings

Before you can customize DB2 Object Comparison Tool 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 Object Comparison Tool 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 2. The Tools Customizer Settings panel (CCQPSET)

- 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 Object Comparison Tool 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 1. 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 GrpAttach 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 2. Value that is used in the CONNECT statements in the generated jobs

| SSID | GrpAttach | 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 parameters. The parameters that are displayed on 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 Object Comparison Tool Discover EXEC is stored. Each product has its own Discover EXEC. The Discover EXEC retrieves the product 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 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 Object Comparison Tool or to change parameter settings.

Related concepts:

“Customizing DB2 Object Comparison Tool”

Using Tools Customizer to customize DB2 Object Comparison Tool consists of identifying the product to customize; defining any required DB2 Object Comparison Tool and DB2 parameters; generating the customization jobs; and submitting the jobs.

Hiding and displaying panel text

After you are familiar with Tools Customizer, you might want to hide the instructions and some of the basic descriptions that are displayed by default on Tools Customizer panels.

About this task

By using the `OPTIONS` command, you can choose to show or hide the following information on Tools Customizer panels:

- The instructions on all panels
- The Product to Customize section on the Customizer Workplace panel (CCQPWRK)
- The Usage Notes section on the Product Parameters panel (CCQPPRD), the LPAR Parameters panel (CCQPLPR), and the DB2 Parameters panel (CCQPDB2).

By hiding this information, more data can be displayed on the panels. Later, you can redisplay this information also by using the `OPTIONS` command.

Procedure

1. On any Tools Customizer panel, issue the `OPTIONS` command. The Panel Display Options panel (CCQPOPT) is displayed, as shown in the following figure. By default, all options are preselected with a slash (/) to be shown.

```
CCQPOPT                Panel Display Options

Select panel display options and press Enter. To cancel, press End.

Panel Display Options
Specify a slash (/) to select options.
/ Show the panel instructions
/ Show the Product to Customize section
/ Show the Usage Notes section
Command ==>
```

Figure 3. The Panel Display Options panel (CCQPOPT)

2. To hide any of the options, remove the slash, and press Enter.

Customizing DB2 Object Comparison Tool

Using Tools Customizer to customize DB2 Object Comparison Tool consists of identifying the product to customize; defining any required DB2 Object Comparison Tool and DB2 parameters; generating the customization jobs; and submitting the jobs.

Customization roadmaps describe the steps that you must complete to customize DB2 Object Comparison Tool. Separate roadmaps are provided for the three most common types of customizations.

Use the following table to determine which roadmap corresponds to your environment.

Table 3. Customization roadmaps

| Environment description | Roadmap |
|--|--|
| You do not have a customized version of DB2 Object Comparison Tool, and you need to customize it for the first time. | "Roadmap: Customizing DB2 Object Comparison Tool for the first time" |
| You have a customized version of of DB2 Object Comparison Tool, but you want to change one or more parameter values. | "Roadmap: Recustomizing DB2 Object Comparison Tool" on page 40 |

Roadmap: Customizing DB2 Object Comparison Tool for the first time

This roadmap lists and describes the steps that are required to customize DB2 Object Comparison Tool for the first time.

Before you complete these steps, ensure that the following prerequisites have been met:

- All of the product customization steps that must be done before Tools Customizer is started are complete.
- Tools Customizer is started.
- The Tools Customizer settings have been reviewed or modified, and saved.

Complete the steps in the following table to customize DB2 Object Comparison Tool for the first time.

Table 4. Steps for customizing DB2 Object Comparison Tool for the first time

| Step | Description | Instructions |
|------|---|---|
| 1 | Specify the product metadata library for the product that you want to customize. The name of this library is <i>hlq.SGOCDENU</i> . | "Specifying the metadata library for the product to customize" on page 41 |
| 2 | Create new DB2 entries and associate them with DB2 Object Comparison Tool. | "Creating and associating DB2 entries" on page 42 |
| 3 | Define the required parameters. | "Defining parameters" on page 44 |
| 4 | Generate the customization jobs for the product or for the DB2 entries on which DB2 Object Comparison Tool is ready to be customized. | "Generating customization jobs" on page 48 |
| 5 | Submit the generated customization jobs. | "Submitting customization jobs" on page 48 |

The following table lists some of the common administrative tasks that you might need to do during the customization process.

Table 5. Administrative tasks

| Description | Instructions |
|---|----------------------------------|
| Browse the different types of parameters. | "Browsing parameters" on page 50 |

Table 5. Administrative tasks (continued)

| Description | Instructions |
|--|---|
| Copy an existing DB2 entry to the list of DB2 entries on which DB2 Object Comparison Tool can be customized. | "Copying DB2 entries" on page 50 |
| Remove one or more DB2 entries from the associated list. | "Removing DB2 entries" on page 52 |
| Delete one or more DB2 entries from the master list. | "Deleting DB2 entries" on page 52 |
| Display a list of customization jobs that have been previously generated. | "Displaying customization jobs" on page 53 |
| Maintain the customization jobs in the customization library. | "Maintaining customization jobs" on page 53 |

Roadmap: Recustomizing DB2 Object Comparison Tool

This roadmap lists and describes the steps to change parameter values and regenerate customization jobs for DB2 Object Comparison Tool after you have customized it for the first time.

The new customization jobs will replace the customization jobs that were previously generated and stored in the customization library. Part of the recustomization process includes selecting or deselecting optional tasks or steps, changing the definitions of parameters that have already been defined, or both. Use the method in this roadmap instead of deleting customization jobs from the customization library.

Before you complete these steps, ensure that the following prerequisites have been met:

- All of the product customization steps that must be done before Tools Customizer is started are complete.
- Tools Customizer is started.

Complete the steps in the following table to recustomize DB2 Object Comparison Tool.

Table 6. Required steps for recustomizing DB2 Object Comparison Tool

| Step | Description | Instructions |
|------|---|---|
| 1 | Specify the product metadata library for the product that you want to recustomize. The name of this library is <i>hlq.SGOCDENU</i> . | "Specifying the metadata library for the product to customize" on page 41 |
| 2 | Edit the specific tasks, steps, or parameters that need to be changed. | <ul style="list-style-type: none"> • "Defining DB2 Object Comparison Tool parameters" on page 45 • "Defining DB2 parameters" on page 46 |
| 3 | Generate the customization jobs for the product or for the DB2 entries on which DB2 Object Comparison Tool is ready to be customized. | "Generating customization jobs" on page 48 |
| 4 | Submit the new generated customization jobs. | "Submitting customization jobs" on page 48 |

The following table lists some of the common administrative tasks that you might need to do during the customization process.

Table 7. Administrative tasks

| Description | Instructions |
|--|---|
| Browse the different types of parameters. | "Browsing parameters" on page 50 |
| Copy an existing DB2 entry to the list of DB2 entries on which DB2 Object Comparison Tool can be customized. | "Copying DB2 entries" on page 50 |
| Remove one or more DB2 entries from the associated list. | "Removing DB2 entries" on page 52 |
| Delete one or more DB2 entries from the master list. | "Deleting DB2 entries" on page 52 |
| Display a list of customization jobs that have been previously generated. | "Displaying customization jobs" on page 53 |
| Maintain the customization jobs in the customization library. | "Maintaining customization jobs" on page 53 |

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 Object Comparison Tool. This information controls what is displayed on the Product Parameters panel and the DB2 Parameters panel.

After DB2 Object Comparison Tool has been SMP/E installed, the default name of the product metadata library is *high_level_qualifier.SGOCDENU*, 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 . DMT00L.GOCA2MPE.SGOCDENU

Previously Used Metadata Library:

=>
=>
=>
=>

```

Figure 4. 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 Object Comparison Tool 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 Object Comparison Tool 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 Object Comparison Tool for the first time**
Do not run the DB2 Object Comparison Tool 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 Object Comparison Tool parameters.

Creating and associating DB2 entries

You can create new DB2 entries and associate them with DB2 Object Comparison Tool.

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:

```
CCQPDA          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

Product to Customize
Product metadata library : DMT00L.GOCA2MPE.SGOCDE      > LPAR . . : 3090
Product name . . . . . : IBM DB2 Object Comparison Tool for z/OS
Product version . . . . . : 11.1.0

Line commands: A - Associate  C - Copy

Cmd SSID GrpAttch
----- End of DB2 entries -----
```

Figure 5. 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:

```
CCQPDCR          Create a DB2 Entry
Command ==>>>

Specify a DB2 subsystem ID, a DB2 group attach name, or both for the
new DB2 entry. Press Enter to continue or End to cancel.

New DB2 Entry Information
DB2 subsystem ID . . . . .
DB2 group attach name . .
```

Figure 6. The Create a DB2 Entry 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 *Inn* 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

Product to Customize
Product metadata library : DMT00L.GOCA2MPE.SGOCDE      > LPAR . . : 3090
Product name . . . . . : IBM DB2 Object Comparison Tool for z/OS
Product version . . . . : 11.1.0

Line commands: A - Associate  C - Copy

Cmd  SSID  GrpAtch
     DB0A  --
----- End of DB2 entries -----

```

Figure 7. The Associate DB2 Entry for Product panel with a new DB2 entry in the master list

- 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 Object Comparison Tool, 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 Object Comparison Tool.

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 234

Tools Customizer uses several unique terms that you should be familiar with before you begin to use Tools Customizer.

Defining parameters

To customize DB2 Object Comparison Tool, you must define DB2 Object Comparison Tool parameters and DB2 parameters, if your customization requires DB2 entries.

About this task

You must define the DB2 Object Comparison Tool parameters first for the following reasons:

- If you ran the DB2 Object Comparison Tool Discover EXEC, you must review the values that were discovered.
- If you select optional tasks and steps on the Product Parameters panel that affect the DB2 entry on which you will customize DB2 Object Comparison Tool, additional parameters might be displayed on the DB2 Parameters panel.

- If other steps must be completed in a specific sequence, customization notes on the Product Parameters panel will display the correct sequence.

Defining DB2 Object Comparison Tool parameters

DB2 Object Comparison Tool parameters are specific to DB2 Object Comparison Tool.

About this task

If you ran the DB2 Object Comparison Tool 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 Object Comparison Tool parameters, a note labeled **Important** will display the correct sequence on this panel.

```

CCQPPRD                               Product Parameters                               11:15:35
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
selected step within a task. Press End to save and exit.

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

Product to Customize
  Product metadata library . : DMT00L.GOCA2MPE.SGOCDE > LPAR . . : 3090
  Product name . . . . . : DB2 Object Comparison > Version . : 11.1.0

Product customization library .: CCQTCZ.SYSADM.CUST.$3090$.GOC1020
More: - +

Create the VB CLIST and EXEC libraries.

/ Create the VB libraries.
DB2 Object Comparison high-level qualifier
                                GOCB10                                >
Fixed to variable blocked VOLSER . . . . USER01
Fixed to variable blocked UNIT . . . . . SYSALLDA
DB2 Admin customization high-level qualifier
                                ADBB10                                >

Sample JCL step for CM Batch

/ Create the CM Batch interface job step.
DB2 Object Comparison high-level qualifier
                                DMT00L.GOCA2MPE                                >

```

Figure 8. 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:

- In the command line, specify the KEYS command, and map EXPAND to one of the function keys.

- 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 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 Object Comparison Tool 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 42.

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:

```

CCQDPDB2                DB2 Parameters                11:18:05
Command ===>                Scroll ===> PAGE

Enter values for all of the DB2 parameters. Press End to save and exit.

Commands: SAVE - Save parameter values

Product to Customize
  Product metadata library . : DMT00L.GOCA2MPE.SGOCDE > LPAR . . . : 3090
  Product name . . . . . : DB2 Object Comparison > Version . . : 11.1.0

DB2 subsystem ID . . . . . : VA1A
DB2 subsystem ID description . . . . . :
Group attach name . . . . . :

General DB2 Information
Mode . . . . . NFM (CM, CM8, CM9, NFM)
Level Number . . . . . 101 (810, 910, 101)

```

Figure 9. The DB2 Parameters panel

2. Specify values for all parameters that are displayed.

Tips:

- In the command line, specify the KEYS command, and map EXPAND to one of the function keys.
- 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 Object Comparison Tool parameters” on page 45
 DB2 Object Comparison Tool parameters are specific to DB2 Object Comparison Tool.

Generating customization jobs

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

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 Object Comparison Tool 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 Object Comparison Tool.

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 2 of 2
Command ==>>>                                     Scroll ==>> PAGE

Submit the members in the order in which they apply to all DB2 entries. To
submit the job, browse the member and issue the TSO SUBMIT command, or browse
the customized library and submit the jobs from there.

Product to Customize
Product metadata library . : GOC.GOCA2MPE.SGOCDENU > LPAR . . . : 3090
Product name . . . . . : DB2 Object Comparison To > Version . . : 11.1.0

Line Commands: E - Edit B - Browse

Product customization library . : CCQTCZ.SYSADM.CUST.$3090$.GOC1020

Cmd Member  SSID GrpAttch Template Date      Description
-----
A0FB2VB    --  --      GOCFB2VB 2013/01/10 Copy the FB libraries to the VB
A1CM2     --  --      GOCCM2  2013/01/10 GOCCM2
-----
End of customized jobs -----

```

Figure 10. 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 Object Comparison Tool assigns the template name.

DB2_entry_ID

Two alphanumeric characters, AA - 99, that Tools Customizer assigns to a DB2 entry.

For example, the XYZBNDDB2_entry_ID_1 and XYZBNDDB2_entry_ID_2 jobs are generated from the XYZBNDGR template, and the XYZ4DB2_entry_ID_1 and XYZ4DB2_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 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 Object Comparison Tool 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 Object Comparison Tool is customized, and the Customizer Workplace panel is displayed. The status is Customized for the DB2 entries on which DB2 Object Comparison Tool 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 Object Comparison Tool.

Browsing parameters

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

Procedure

1. On the Customizer Workplace panel, specify B next to the **Product 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 *Imm* line command, where *mm* 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 234

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 234

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 Object Comparison Tool 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 Object Comparison Tool 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 Object Comparison Tool 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: Prepare ADBL CLIST

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

The ADBL CLIST brings up the DB2 Admin Main Menu. By using the PANEL(GOCMENU) parameter, you may instead bring up the DB2 Object Compare main menu.

You can start 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.

Before you can use DB2 Object Comparison Tool, you must first allocate the libraries to your ISPF session. Three methods are available to allocate libraries.

Choose one of the following three methods that is most appropriate for your installation to allocate the libraries to your ISPF session:

- Use the PRODADD and LIBAPRE parameters on the ADBL CLIST to specify the unique library names for DB2 Object Comparison Tool libraries. If you are currently using the ADBL CLIST to LIBDEF (allocate) the Administration Tool ISPF libraries, you should also use the ADBL CLIST to LIBDEF DB2 Object Comparison Tool libraries, as shown in the following example:

```
TSO %ADBL PRODADD(GOCB10) LIBAPRE(SGOC)
```

- If your installation copied the DB2 Administration Tool ISPF libraries to a set of libraries that are allocated before you start ISPF, you can copy DB2 Object Comparison Tool ISPF libraries into these same libraries, or you can allocate additional ISPF data sets.
- If you have a personal set of ISPF libraries, you can copy the DB2 Object Comparison Tool ISPF libraries to these data sets. To verify that you have allocated (performed a LIBDEF) the correct ISPF libraries, you can use the ISPF command ISPLIBD. You can also use the TSO ALTLIB DISPLAY command to verify the CLIST and EXEC library allocations.

A compare job can be run either in batch or online. Compare jobs that run online require access to the DSNHDECP module and access is available only if the DB2 libraries are set up properly. If the DB2 load library data set does not exist in the system LINKLIST, the data set must be added to the STEPLIB of the TSO logon procedure. If the DB2 load library data set does not exist in the system LINKLIST or in the STEPLIB, the following error is returned in the compare output when an online compare is run:

```
Unable to load DB2 DECP module: rc = 8. Compare function is terminated.
```

Optional: Editing the GOCFB2VB job

The GOCFB2VB job creates variable-blocked (VB) versions of the DB2 Admin and Object Comparison CLIST and EXEC libraries. You might need to edit the job to specify the correct ADB and GOC SAMP datasets.

Before you begin

The Tools Customizer has generated the GOCFB2VB job.

About this task

Open the GOCFB2VB job in the ISPF editor.

Procedure

1. Edit the job step that creates the VB version of the CLIST library.

Here is an example of a job step.

```
//*****  
//*  
//CLIST EXEC PGM=IKJEFT01,  
// PARM=('%ADBFVBV GOCB10.SGOCCLST',  
//      'GOCB10.SGOCCLST.VB')  
//SYSEXEC DD DISP=SHR,DSN=ADBB10.SADBSAMP  
//SYSTSPRT DD SYSOUT=*  
//SYSTSIN DD DUMMY  
//SYSPRINT DD SYSOUT=*  
//SYSIN DD DISP=SHR,  
//      DSN=ADBB10.SADBSAMP(ADBIEBVB)  
//MEMBERS DD *  
*  
//*
```

- a. Specify the low-level qualifier. In this example, if SGOCCLST is not the low-level qualifier, type the correct one. The job creates the GOCB10.*low-level qualifier*.VB dataset, where *low-level qualifier* is the low-level qualifier that you specify.
2. Edit the job step that creates the VB version of the EXEC library.

Here is an example of a job step.

```
//*****  
//*  
//EXEC EXEC PGM=IKJEFT01,  
// PARM=('%ADBFVBV GOCB10.SGOCEXEC',  
//      'GOCB10.SGOCEXEC.VB')  
//SYSEXEC DD DISP=SHR,DSN=ADBB10.SADBSAMP  
//SYSTSPRT DD SYSOUT=*  
//SYSTSIN DD DUMMY  
//SYSPRINT DD SYSOUT=*  
//SYSIN DD DISP=SHR,  
//      DSN=ADBB10.SADBSAMP(ADBIEBVB)  
//MEMBERS DD *  
*  
//*
```

- a. Specify the low-level qualifier. In this example, if SGOCEXEC is not the low-level qualifier, type the correct one. The job creates the GOCB10.*low-level qualifier*.VB dataset, where *low-level qualifier* is the low-level qualifier that you specify.
3. Save the file.

What to do next

Submit the job.

Optional: Customizing JCL

You might need to customize the DB2 Object Comparison Tool JCL to adhere to your installation standards. Most other skeletons will not require configuration. If necessary, you can configure the JCL that is used by DB2 Object Comparison Tool to run DB2 utilities and other DB2 functions by modifying the skeletons in the SADBSLIB and SGOCSLIB libraries.

About this task

Because member ADBAPY uses ISPF batch for its generated apply job, its skeleton might require more extensive configuration than the other skeletons.

Members of SADBSLIB that might require configuration are:

ADBAPY

Generate apply job/step (uses ISPF batch)

ADBDCMD

Execute DB2 commands

ADBEDDL

Execute DDL files (DROP, CREATE, ALTER)

ADBTCHK

Generate check data job/step

ADBTHPU

Generate High Performance Unload job/step

ADBTIMC

Generate image copy job/step

ADBTREL

Generate (RE)LOAD job/step

ADBTREO

Generate REORG job/step

ADBTREN

Generate RUNSTATS job/step

ADBTUNL

Generate UNLOAD job/step

ADBS27AC

Generate convert job/step

Members of SGOCSLIB that might require configuration are:

GOCCMP

Generate compare job/step

GOCSDB2

Generate extractions from DB2 catalog for source or target

GOCSDDL

Generate extractions from DDL for source or target

Optional: Conforming DB2 Administration Tool data set names

You can align DB2 Administration Tool data set names with your local data set naming conventions.

About this task

To align data sets, use the following steps:

1. Modify the ADB2UCUS skeleton that resides in the ISPSLIB library.
When you subsequently run SMP/E to receive and apply SMP/E usermod ADBU002, the updated ISPF JCL skeletons are added to the SADBSLIB library.

2. For testing purposes, copy the ADB2UCUS skeleton to a private skeleton library and make your changes.

This private skeleton library must be allocated first in the ISPSLIB concatenation (using the USERADD parameter of the ADBL CLIST).

3. After testing is complete, use an SMP/E USERMOD to update the DB2 Admin Tool product libraries.

A sample SMP/E USERMOD is provided in member ADBU002 in the SADBSSAMP library. Instructions for completing this step are provided in sample job ADBU002.

You can use the following variables (only a partial list). A complete list is located in SLIB member ADB2UCUT.

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 configuration are preceded by SET statements which are indicated by)SET.

Example: This example demonstrates several different types of changes to 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
```

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.

Performance considerations

The performance of DB2 Object Comparison Tool can be impacted by several factors.

For optimal performance, avoid these situations:

- Large lists of translation masks
- Comparisons of many objects (especially many views)
- A large number of changes

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*.

Chapter 4. Creating DB2 Object Comparison Tool batch jobs

The process of comparing DB2 objects consists of creating a batch job in which you specify all the aspects of the comparison, and then you run the job. You can create batch jobs for your comparison and save them for later retrieval by using the Object comparison Tool panels. The batch job is generated based on the values that you supplied for options 1 through 5 on the DB2 Object Comparison Tool Menu.

```
Compare ----- DB2 Object Comparison Tool Menu ----- 09:38
Option ==>

      1 - Specify compare source (new)           Status:
      2 - Specify compare target (old)         DDL file specified
      3 - Specify compare masks                DB2 catalog extract specified
      4 - Specify fields to ignore             None specified
      5 - Generate compare job                 Using defaults
                                           Generated

W - Walk through steps 1 - 5 in sequence
V - Generate job to extract version file from source only

R - Reset all
RS - Reset source
RT - Reset target

S - Save dialog
M - Manage/Restore dialog
MC- MultiCompare
MR- Manage saved compare results
```

Figure 11. DB2 Object Comparison Tool Menu (GOCMENU)

The following topics provide additional information:

- “DB2 Object Comparison Tool Menu options” on page 62
- “Specifying source object definitions to be compared” on page 64
- “Specifying target objects to be compared” on page 88
- “Specifying the compare masks data set” on page 90
- “Specifying compare ignore fields” on page 95
- “Specifying that the source is a version scope” on page 80
- “Generating a compare batch job” on page 105
- “Saving dialogs” on page 138
- “Managing and restoring dialogs” on page 139
- “Comparing multiple sources and targets” on page 140

Select options 1 through 4 on the DB2 Object Comparison Tool Menu to specify attributes of the DB2 objects to compare. Then use option 5 to specify the parameters to use for the batch compare job and generate an Object Comparison Tool batch job based on the values that you supplied. You must specify a source and at least one target object before using option 5 to generate a job.

Alternatively, you can use the W option (Walk through steps 1 – 5 in sequence) to proceed directly to each step in options 1 through 5 in succession without returning to the DB2 Object Comparison Tool Menu and selecting the next option manually. When using the W option, the final panel for the current option contains

a Continue command that you can use to display the next panel in the sequence rather than returning to the DB2 Object Comparison Tool Menu.

The Status field indicates the selections that you have made. Press PF1 to display online help for the menu.

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.

Requirement: If you plan to request the generation of apply jobs during the Compare dialog, you must run the Compare dialog connected to the target DB2 system to pick up the correct libraries for use in the apply jobs. Alternatively, if the dialog is run on a different DB2 system, you must manually update the apply jobs to use the correct DB2 libraries.

DB2 Object Comparison Tool Menu options

Use the DB2 Object Comparison Tool Menu options to specify the criteria for the comparison that you want to run.

The DB2 Object Comparison Tool Menu options are:

1 - Specify compare source (new)

Select this option to begin specifying the DB2 source objects to be compared.

“Specifying source object definitions to be compared” on page 64 describes these panels.

2 - Specify compare target (old)

Select this option to begin specifying the DB2 target objects to be compared.

“Specifying target objects to be compared” on page 88 describes the panels that are associated with this task.

3 - Specify compare masks

Select this option to specify that names and qualifiers are to be translated by using masks before the comparison is performed.

“Specifying the compare masks data set” on page 90 describes the panels that are associated with this task.

4 - Specify fields to ignore

Select this option to specify that certain fields should be ignored when the comparison is performed.

“Specifying compare ignore fields” on page 95 describes the panels that are associated with this task.

5 - Generate compare job

Select this option to generate the batch compare job.

“Generating a compare batch job” on page 105 describes these panels.

W - Walk through steps 1 – 5 in sequence

Select this option to proceed directly to each step in succession without returning to the DB2 Object Comparison Tool Menu and selecting them

individually. See Chapter 4, “Creating DB2 Object Comparison Tool batch jobs,” on page 61 for more information about this process.

V - Generate job to extract version file from source only

Select this option to generate a batch job that creates a version file from the source only. The version file can be used for subsequent compare operations, using this version as the source or target. This option can be used to:

- Create a version file on one system
- Transfer the version file to another system
- Generate a compare job on the other system

R - Reset status

Select this option to clear the Status fields and enter your specifications again.

S - Save dialog

Select this option to store the current selections for later retrieval and subsequent reuse. Refer to “Saving dialogs” on page 138 for information about how to save a dialog.

M - Manage/Restore dialog

Select this option to retrieve, rename, or delete a previously saved dialog. Refer to “Managing and restoring dialogs” on page 139 for information about how to use the Manage and Restore dialog functions.

MC - MultiCompare

Select this option to compare one or more saved dialogs. Refer to “Comparing multiple sources and targets” on page 140 for information about comparing multiple objects.

MR - Manage saved compare results

Select this option to manage and view the saved compare results.

The Version File Conversion utility

Use the Version File Conversion utility to convert a file from one version to the highest level of DB2 that is supported by the current version of DB2 Admin tool and DB2 Object Comparison Tool.

Each version of the DB2 Admin tool and DB2 Object Comparison Tool establishes a range of supported version file levels. The Version File Conversion utility converts a version file within the established range to the current level before it is processed. In contrast, if an unconverted version file that is within the established range is passed to DB2 Object Comparison Tool, the tool can process the version file but requires more conversion time. The Version File Conversion utility also converts the version file catalog records to the highest level of DB2 that is supported by the current version of DB2 Admin tool and DB2 Object Comparison Tool. The original version file is unchanged by DB2 Object Comparison Tool.

If the version file is within the supported range, the Version File Conversion Tool can permanently upgrade the file to the current supported level. A converted version file can be used by DB2 Object Comparison Tool immediately, saving the time that the conversion would require.

The Version File Conversion tool is started using a batch job. Sample JCL is provided in the SAMP library ADBVFCO member. The JCL must be tailored to your installation before use.

The tool can be used to convert version files that are stored in sequential data sets, PDS data sets, and PDSE data sets. It can also convert base versions that are stored using DB2 Admin Tool.

Input to the Version File Conversion Tool is specified using the VFLIST DD statement. For a sequential to sequential data set conversion, specify the existing version file with the VFOLD keyword and the new version file with the VFNEW keyword. For example:

```
VFOLD='C386799.DT27760.C.VERSION(SOURCE00)', VFNEW='C386799.DT27760.C.SOURCE00';  
VFOLD='Z9.ORDER.VERSION(MAY29)', 'Z9.ORDER.VERSION(JUNE17)';
```

The sequential data sets named must be allocated and cataloged first. If VFOLD or VFNEW are PDS data sets, then the member names must be included. The VFNEW PDS member name can be a new name; however, the VFOLD and VFNEW cannot have the same name. The data set must not contain any sequence numbers in columns 72 - 80.

You can convert a base version file in one of two ways: VOWNER and VNAME or by VID. Change Management option 4.1 lists all base version files that are stored in DB2 Admin. VOWNER, VNAME, and VID are three of the fields that are listed for each base version. A base version file can be described by VOWNER='owner_name', VNAME='version_file_name'; or by VID=nnn. The converted base version replaces the original version only if no errors are detected.

A base version file can be converted and stored in a sequential, PDS, or PDSE data set. To convert a file, specify the base version file using VOWNER and VNAME or VID and specify the output data set using VFNEW. DB2 Object Comparison Tool converts the base version file to the current level and writes it to the data set specified by VFNEW. The original base version file remains unchanged.

The Version File Conversion tool processes each version file sequentially. If an error is detected, an error message is issued and processing is halted for the version file in error. The next version file is then processed.

Specifying source object definitions to be compared

The first step in defining the attributes of the DB2 objects to compare is to specify if the source object is from a DDL file, from the DB2 catalog, or from a compare version file. Use the Specify Compare Source panel to define these attributes.

About this task

Use the following steps to specify the source object definitions:

1. Select option 1 from the DB2 Object Comparison Tool Menu to display the Specify Compare Source panel:


```

GOC1 re ----- Specify Compare Source ----- 17:01
Option ==> VS

1 - Source is from a DDL file
2 - Source is from the DB2 catalog
3 - Source is from a compare version file

VS - Source is from the DB2 catalog and the objects are selected from
    a version scope

Exclude Specification:
Owner . . . . . > (Optional, default is ELACZ, ? to lookup)
Name . . . . . > (Required, ? to lookup)
Edit objects . . . . YES (Yes/No)

```

Figure 12. Specify Compare Source panel (GOC1)

2. Specify the source definition for the DB2 objects that will be used in the compare job that you are creating:
 - Select option 1 to specify that the definition of the source DB2 objects is from a DDL file, that is, a file that contains SQL CREATE statements.
“Specifying that the source is a DDL file” describes the panels that are associated with this option.
 - Select option 2 to specify that the definition of the source DB2 objects is from an extract from the DB2 catalog, that is, an extract of one or more databases, table spaces, or tables and all the dependent objects.
“Specifying that the source is from an extract from the DB2 catalog” on page 67 describes the panels that are associated with this option.
 - Select option 3 to specify that the definition of the source of DB2 objects is from a previously created version file.
“Specifying the input version file data set name” on page 79 describes the panels that are associated with this option.
 - Select option VS to specify that the source is from the DB2 catalog and the objects are selected from a version scope.
“Specifying that the source is a version scope” on page 80 describes the panels that are associated with this option.
3. Enter the values for the subsequent panels and press Enter to return to the DB2 Object Comparison Tool Menu.

Specifying that the source is a DDL file

Select the Specifying source DDL file definitions option to specify the data set name of the DDL file that contains the objects to compare, the data set name of the version file in which you want to put the definition of these objects, and an optional description of the objects that you are comparing.

About this task

To specify source DDL file definitions:

1. Select option 1 on the Specify Compare Source panel.
If Change Management is enabled on your system, the Specify Source DDL panel, as shown in the following figure, is displayed. If Change Management is not enabled, the Version table Owner and Name fields are not available on this panel.

```

Compare ----- Specify DDL File -----
Option ==>

                                                                    More:  +

Specify input DDL file:
  DDL data set . .

Specify compare version file output:
  Version table entry:
    Owner . . .                (? to look up)
    Name . . .                 (? to look up)
  Data set:
    Data set name .

Enter a description (optional):
  Description . .

----- DB2 Object Compare Warning -----
  You have asked to generate a version file, but the source for these
  objects is a DDL file.  If this DDL does not include all dependent objects
  and authorizations, any subsequent comparison with this version file
  may lead to loss of these dependent objects.  Press ENTER to continue
  or END to stop this operation.
                                     Please read this carefully
-----

```

Figure 13. Specify Source DDL File panel (GOC11)

2. In the DDL data set field, specify the name of the data set that contains the SQL that defines the objects to compare. The data set must contain valid SQL statements and must adhere to TSO naming conventions. The data set can be either of these types:
 - 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)

The SQL statements that define the objects must be in columns 1-72 of the data set.

Elements of a DDL statement can span records in the data set. Column 1 of a record is considered to immediately follow column 72 of the previous record. This convention can be used for long names or long string constants.

3. If you are using Change Management, specify the version table owner and name. Enter a question mark (?) in the field to display the CM Versions panel (ADB2C41) from which you can select a version file from the Change Management database.

Attention: If you select the option to store the version file in the database, an additional step is created in the compare job to store the version file for both the source and target objects in the database. If both the database and the data set are specified, the information in the data set field is used to determine the output destination for the version file.

Refer to “Managing changes to DB2 objects” in *IBM DB2 Administration Tool for z/OS User’s Guide and Reference* for additional information about Change Management.

4. Alternatively, if you are not using Change Management, specify the name of the output data set for the version file in the Data set name field. The Object Comparison Tool DDL extract program extracts the definitions and puts them into this data set; this data set becomes input to the compare process. If an existing data set is specified, it is overwritten. The specified data set must be one of the following types:
 - A variable-block sequential data set

- A member of a partitioned data set with a record length of 16K (RECFM=Vx,LRECL=16384)

If the specified version data set does not exist, it is created.

5. Optionally, in the Description field, specify a description of the DB2 objects that you are comparing. The description is printed in the Comparison report, is placed in the header record of the version file, and is used to describe the common properties of the DB2 objects. Any text that you enter is shown in the compare reports.

Example: APPLICATION V2

6. Press Enter to return to the DB2 Object Comparison Tool Menu.

Attention: If you attempt to request a version file that was created from DDL and the DDL does not contain all the dependent objects and relations (referential integrity), the result could be the loss of such objects and relations. The warning message shown in the Specify Source DDL File panel is displayed when this condition occurs. You have the option to continue or end the operation.

Related concepts:

Chapter 5, “Batch DDL file extraction program,” on page 147

The DDL file extraction program interprets a source file of DDL statements that define DB2 objects. The program generates an output file, called a *version file*, that contains records that are similar in format to those in the DB2 catalog that defines the same objects.

Specifying that the source is from an extract from the DB2 catalog

Use the Specify DB2 Source Catalog Extract panel to specify the DB2 catalog extract option, a version file output data set, and an optional description of the objects that you are comparing.

About this task

When you extract databases, the databases and all their dependent objects are extracted, which means that the databases, table spaces, tables, views, indexes, aliases, synonyms, and so forth are extracted.

If the level of the databases is too high, you can extract a table space from the DB2 catalog, and then the databases are not extracted. If you specify each table space, all dependent objects are extracted. The same is true for tables.

To specify source extract definitions:

1. Select option 2 on the Specify Compare Source panel.

If Change Management is enabled on your system, the Specify DB2 Source Catalog Extract panel, as shown in the following figure, is displayed. If Change Management is not enabled, the Version table Owner and Name fields are not available on this panel.

```

Compare ----- Specify DB2 Catalog Extract ----- 11:08
Option ==>

  1 - Source is databases from the DB2 catalog
  2 - Source is table spaces from the DB2 catalog
  3 - Source is tables from the DB2 catalog
  4 - Add schema objects to the DB2 Source catalog extract

Specify compare version file output:
Version table entry:
  Owner . . .                (? to look up)
  Name . . .                 (? to look up)
Data set:
  Data set name . .

Enter a description (optional):

  Description . .

```

Figure 14. Specify DB2 Source Catalog Extract panel (GOC12)

2. If you are using Change Management, specify the version table owner and name. Enter a question mark (?) in the field to display the CM Versions panel (ADB2C41) from which you can select a version file from the Change Management database.

Attention: If you select the option to store the version file in the database, an additional step is created in the compare job to store the version file for both the source and target objects in the database. If both the database and the data set are specified, the information in the data set field is used to determine the output destination for the version file.

Refer to “Managing changes to DB2 objects” in *IBM DB2 Administration Tool for z/OS User’s Guide and Reference* for additional information about Change Management.
3. Alternatively, if you are not using Change Management, in the version data set field, specify a version data set name. Specify the name of the output data set for the version file. The compare DB2 Catalog extract program extracts the definitions and puts them into this data set; this data set becomes input to the compare process. If an existing data set is specified, it is overwritten. The specified data set must be one of the following types:
 - A variable-block sequential data set
 - A member of a partitioned data set with a record length of 16K (RECFM=Vx, LRECL=16384)

If the specified version data set does not exist, it is created.
4. Optionally, in the Description field, specify a description of the DB2 objects that you are comparing. The description is placed in the header record of the version file and can be used to describe the common properties of the DB2 objects. Any text you enter is shown in the compare reports.

Example: APPLICATION V2
5. Specify the source:
 - Select option 1 to find and select the databases to use as input to the compare process.

“Specifying that the source input is databases from the DB2 catalog” on page 69 describes these panels.
 - Select option 2 to find and select the table spaces to use as input to the compare process.

“Specifying that the source input is DB2 table spaces” on page 72 describes these panels.

- Select option 3 to find and select the tables to use as input to the compare process.

“Specifying that the source input is DB2 tables” on page 75 describes these panels.

- Select option 4 to extract schema-based objects (user-defined functions, user-defined types, stored procedures, and sequences).

“Specifying the source input DB2 schema” on page 77 describes these panels.

6. Press Enter.

The panel that corresponds to the source option that you selected is displayed.

Related tasks:

“Specifying that the source input is databases from the DB2 catalog”

Use the Specify Source DB2 Databases panel to find and select the databases to use as input to the compare process.

“Specifying that the source input is DB2 table spaces” on page 72

Use the Specify Source DB2 Table Spaces panel to specify the source DB2 table spaces to be used as input to the compare process. The source objects are DB2 tables and all of their dependent objects.

“Specifying that the source input is DB2 tables” on page 75

Use the Specify Source DB2 Tables panel to specify the source DB2 tables to be used as input to the compare process.

“Specifying the input version file data set name” on page 79

Select the Specifying the input version file data set name option to specify the data set name of the version file that is to be used as input to the compare process (source or target).

Specifying that the source input is databases from the DB2 catalog

Use the Specify Source DB2 Databases panel to find and select the databases to use as input to the compare process.

About this task

To specify that the source input is databases from the DB2 catalog:

1. Select option 1 on the Specify DB2 Source Catalog Extract panel.

The Specify Source DB2 Databases panel is displayed.

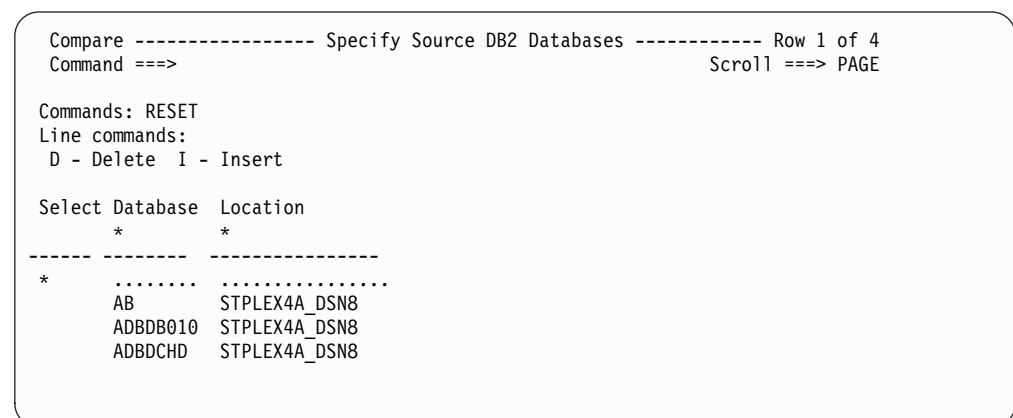


Figure 15. Specify Source DB2 Databases panel (GOC1D)

The list of databases is empty the first time you display Specify Source DB2 Databases panel.

The row that contains “.....” is used only for line commands.

Issue the RESET command to remove all objects from the list.

2. Specify the name of the source DB2 database.
Use the line commands that are shown on the panel to modify the list of objects.
3. Specify a unique location name for an accessible server. All selected objects must be from the same location (you cannot compare objects from different locations).
4. Press PF3 to return to the DB2 Object Comparison Tool Menu or issue the Insert line command (I) to display the Compare Add Databases panel.

```
Compare ----- DSN8 Compare Add Databases ----- 14:00
Option ==>

Enter the partial name of the database you want to add to the compare
operation:

Partial database name . . . : DS%
Location name . . . . . :

Press enter to search for the database.
```

Figure 16. DB2 Compare Add Database panel (GOC1DA)

Use the DB2 Compare Add Databases panel to specify a partial database name as a DB2 catalog search criteria for the databases to add to the compare process.

5. Specify a partial database name by using the wildcard character (%). When a wildcard is used, Object Comparison Tool searches the DB2 catalog for databases that match the partial name and any values in the place of the wildcard.
6. Specify a unique location name for an accessible server. If you do not specify a location name, the location name of the current server is used.

The DB2 Compare Add Databases panel, shows the DB2 databases that match the partial database name specified in DB2 Compare Add Database panel.

```

Compare ----- DSN8 Compare Add Databases ----- Row 1 of 24
Command ==>>>                                     Scroll ==>> PAGE

Valid line commands are:                            Location: STPLEX4A_DSN8
S - Select (add)

Select Database Action
*          *
-----
S      DSG24D0G
      DSG24D0X
S      DSG24D1Z
      DSNAE71A
      DSNAE71P
      DSNATPDB
      DSNDB04
      DSNDB06
      DSNDB07
      DSNDPSM
      DSNRGFDB
      DSNRLST
      DSNRTSDB
      DSN8CDDDB
      DSN8D71L
      DSN8TEMP

```

Figure 17. DB2 Compare Add Databases panel 1 (GOC1DD)

The Compare Add Databases panel contains the following fields:

Database

This column shows all of the databases that match the partial name you specified in the DB2 Add Database panel.

Action

This field shows the action that was performed for the corresponding database.

Location

This field indicates the unique location name for an accessible server. If you did not specify a location name, the location name of the current server was used.

- Issue the Select command in the Select field that corresponds to the database to add to the list of databases to be compared or issue the ALL command to select all objects in the list.

The DB2 Compare Add Databases panel shows the result of any actions you took against the databases on the Compare Add Databases panel.

```

Compare ----- DB2 Compare Add Databases ----- Row 1 of 16
Command ==>                                     Scroll ==> PAGE

Valid line commands are:                          Location: STPLEX4A_DSN8
S - Select (add)

Select Database Action
      *          *
-----
*      DSG24D0G Added
      DSG24D0X
      DSG24D1Z Added
      DSNAE71A
      DSNAE71P
      DSNATPDB
      DSNDB04
      DSNDB06
      DSNDB07
      DSNDBSM
      DSNRGFDB
      DSNRLST
      DSNRTSDB
      DSN8CDDB
      DSN8D71L
      DSN8TEMP

```

Figure 18. DB2 Compare Add Databases panel 2 (GOC1DD)

In this case, databases DSG24D0G and DSG24D1Z were added to the list of databases to be compared.

Related tasks:

“Specifying that the source is from an extract from the DB2 catalog” on page 67 Use the Specify DB2 Source Catalog Extract panel to specify the DB2 catalog extract option, a version file output data set, and an optional description of the objects that you are comparing.

Specifying that the source input is DB2 table spaces

Use the Specify Source DB2 Table Spaces panel to specify the source DB2 table spaces to be used as input to the compare process. The source objects are DB2 tables and all of their dependent objects.

About this task

To specify that the source input is table spaces:

1. Select option 2 on the Specify DB2 Source Catalog Extract panel.
The Specify Source DB2 Table Spaces panel is displayed.


```

Compare ----- Specify Source DB2 Table Spaces ----- Row 1 of 1
Command ==>                                         Scroll ==> PAGE

Commands: RESET
Line commands:
D - Delete I - Insert

          Table
Select Database Space      Location
          *      *      *
-----
          .....

```

Figure 19. Specify Source DB2 Table Spaces panel (GOC1S)

The list of table spaces is empty the first time that you display the Specify Source DB2 Table Spaces panel.

The row that contains “.....” is used only for line commands. Use the line commands shown on the panel to modify the list of objects.

Issue the RESET command to remove all objects from the list.

The fields on the Specify Source DB2 Table Spaces panel are:

Database
This column shows the list of databases resulting from a query run from the Compare Add Table Spaces panel. Refer to Figure 19.

Table Space
This column shows the list table spaces resulting from a query run from the Compare Add Table Spaces panel. Refer to Figure 19.

Location
Shows the corresponding database location name for an accessible server. All selected objects must be from the same location (you cannot compare objects from different locations).

- Issue the Insert line command (I) in the Select column.
The Compare Add Table Spaces panel is displayed.

```

Compare ----- DSN8 Compare Add Table Spaces ----- 14:03
Option ==>

Enter the partial name of the table space you want to add to the compare
operation:

Partial database name . . : DS%

Partial table space name . : F%

Location name . . . . . :

Press Enter to search for the table spaces.

```

Figure 20. Compare Add Table Spaces panel (GOC1SA)

3. Specify a partial database name by using the wildcard character (%). When a wildcard is used, Object Comparison Tool searches the DB2 catalog for databases that match the partial name and any values in the place of the wildcard.
4. Specify a partial table space name by using the wildcard character (%). When a wildcard is used, Object Comparison Tool searches the DB2 catalog for table spaces that match the partial name and any values in the place of the wildcard.
5. Specify a unique location name for an accessible server and press Enter. If you do not specify a location name, the location name of the current server is used. The Compare Add Table Spaces panel shown in the following figure is displayed.

```

Compare ----- DSN8 Compare Add Table Spaces ----- Row 1 of 11
Command ==>>                                         Scroll ==>> PAGE

Valid line commands are:                               Location: STPLEX4A_DSN8
S - Select (add)

Select Database Space Action
-----
DSNDB04  FACTLEVE
DSNDB04  FEDEXP
DSNDB04  FEDTEST
DSNDB04  FEST1234
DSNDB04  FIRSTTS1
DSNDB04  FLOAT
DSNDB04  FRED
DSNDB04  FRED2
DSNDB04  FSRCOR
DSNDB04  FSUM2
DSNDB04  FSUM4

```

Figure 21. Compare Add Table Spaces panel (GOC1SD)

This panel shows the table spaces that match the partial table space name (F%) that you specified on the preceding panel.

The Compare Add Table Spaces panel contains the following fields:

Database

This column shows all of the databases that match the partial name you specified in the DB2 Compare Add Database panel.

Table Space

This column shows all of the table spaces that match the partial name you specified in the DB2 Compare Add Database panel.

Action

This field shows the action that was performed for the corresponding database.

Location

This field indicates the unique location name for an accessible server. If you did not specify a location name, the location name of the current server was used.

6. Issue the Select command in the Select field that corresponds to the table space to add to the list of tables spaces to be compared or issue the ALL command to select all objects in the list.

Related tasks:

“Specifying that the source is from an extract from the DB2 catalog” on page 67 Use the Specify DB2 Source Catalog Extract panel to specify the DB2 catalog extract option, a version file output data set, and an optional description of the objects that you are comparing.

Specifying that the source input is DB2 tables

Use the Specify Source DB2 Tables panel to specify the source DB2 tables to be used as input to the compare process.

About this task

To specify that the source input is a table:

1. Select option 3 on the Specify DB2 Source Catalog Extract panel.

The Specify Source DB2 Tables panel is displayed.

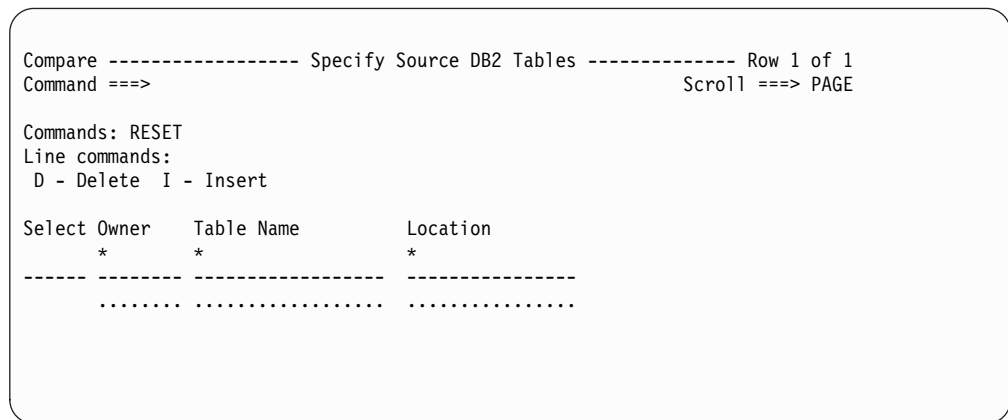


Figure 22. Specify Source DB2 Tables panel (GOC1T)

The list of tables is empty the first time that you display this panel.

The row that contains “.....” is used only for line commands. Use the line commands that are shown on the panel to modify the list of objects.

The Specify Source DB2 Tables panel contains the following fields:

Owner

This column shows the list of database table owners resulting from a query run from the Compare Add Table panel. Refer to Figure 22.

Table Name

This column shows the list of table names resulting from a query run from the Compare Add Table panel. Refer to Figure 22.

Location

This column shows the corresponding database location name for an accessible server. All selected objects must be from the same location (you cannot compare objects from different locations).

2. Issue the RESET command to remove all objects from the list.
3. To continue, issue the Insert line command (I) in the Select column.

The Compare Add Tables panel shown in the following figure is displayed. Use the Compare Add Tables panel to specify a partial table name as a DB2 catalog search criteria for the tables to add to the compare process.

```

Compare ----- DSN8 Compare Add Tables ----- 14:05
Option ==>

Enter the partial owner and name of the table you want to add to the compare
operation:

Partial table owner . . . : RAJESHR

Partial table name . . . . : MY%

Location name . . . . . :

Press Enter to search for the tables.

```

Figure 23. Compare Add Tables panel 1 (GOC1TA)

4. Specify a partial table owner by using the wildcard character (%). When a wildcard is used, Object Comparison Tool searches the DB2 catalog for tables that match the partial owner name and any values in the place of the wildcard.
5. Specify a partial table name by using the wildcard character (%). When a wildcard is used, Object Comparison Tool searches the DB2 catalog for tables that match the partial name and any values in the place of the wildcard.
6. Specify a unique location name for an accessible server. If you do not specify a location name, the location name of the current server is used.
7. Press Enter.

The Compare Add Tables panel shows the DB2 tables that match the partial table name that you specified in the preceding panel:

```

Compare ----- DSN8 Compare Add Tables ----- Row 1 of 6
Command ==>                                     Scroll ==> PAGE

Valid line commands are:                          Location: STPLEX4A_DSN8
S - Select (add)

Select Owner      Table Name      Action
   *              *                *
-----
RAJESHR MYTEST
RAJESHR MYINVOICE
RAJESHR MYINV2
RAJESHR MYINV3
RAJESHR MYSPECS
RAJESHR MYSPECS1

```

Figure 24. Compare Add Tables panel (GOC1TD)

The Compare Add Tables panel contains the following fields:

Owner

This column shows all table owner names that match the partial name you specified in the Compare Add Tables panel.

Table Name

This column shows all table names that match the partial name you specified in the Compare Add Tables panel.

Action

This field shows the action that was performed for the corresponding database.

Location

This field indicates a unique location name for an accessible server. If the name was not specified, the location name of the current server is used.

8. Issue the Select command in the Select field that corresponds to the table to add to the list of tables to be compared or issue the ALL command to select all objects in the list.

Related tasks:

“Specifying that the source is from an extract from the DB2 catalog” on page 67
Use the Specify DB2 Source Catalog Extract panel to specify the DB2 catalog extract option, a version file output data set, and an optional description of the objects that you are comparing.

Specifying the source input DB2 schema

Use the Specify Source DB2 Schema panel to specify schema-based objects to be extracted. Schema-based objects are user-defined functions, user-defined types, stored procedures, and sequences.

About this task

Requirement: You must select schema-based objects from the same location as previously selected objects.

After the extraction is complete, you can add schema objects. You can extract user-defined functions, stored procedures, native SQL procedures, and distinct types. If you also have a trigger as a schema-defined object, it is extracted whenever you extract a table where a trigger is defined. Therefore, it is not necessary to define a trigger as a separate object.

You can choose this option separately or as an additional option (you can combine this option with option 1, 2, or 3 or use it by itself).

Use the following steps to add schema objects to the DB2 source catalog extract:

1. Select option 4 on the Specify DB2 Source Catalog Extract panel (Figure 14 on page 68).
The Specify Source DB2 Schema panel is displayed.

```

Compare ----- Specify Source DB2 Schema ----- Row 1 of 4
Command ==>                                         Scroll ==> PAGE

Commands: Reset
Line commands:
D - Delete I - Insert

      Schema
Select Name      Location
      *          *
-----
      .....

```

Figure 25. Specify Source DB2 Schema panel (GOC1C)

The list of schemas is empty the first time that you display the Specify Source DB2 Schema panel.

The row that contains “.....” is used only for line commands. Use the line commands shown on the panel to modify the list of objects.

The Specify Source DB2 Schema panel contains the following fields:

Schema Name

This column shows the list of schema names resulting from a query run from the Compare Add Table Spaces panel.

Location

This column shows the corresponding database location name for an accessible server. All selected objects must be from the same location (you cannot compare objects from different locations).

2. Issue the RESET command to remove all objects from the list.
3. Issue the Insert line command (I) in the Select column.

The Use the Compare Add Schema Panel is displayed.

```

Compare ----- DSN8 Compare Add Schema ----- 14:08
Option ==>

Enter the partial name of the schema you want to add to the compare operation:

Partial schema name . . . : R%
Location name . . . . . :

Press Enter to search for the schema.

```

Figure 26. Compare Add DB2 Schema panel (GOC1CA)

Use the Compare Add Schema panel to specify a partial schema name as a DB2 catalog search criteria for the schema to add to the compare process.

4. Specify a partial schema name by using the wildcard character (%). When a wildcard is used, Object Comparison Tool searches the DB2 catalog for databases that match the partial name and any values in the place of the wildcard.

In the previous figure a partial schema name (R%) is entered with a wildcard.

- Specify a unique location name for an accessible server. If you do not specify a location name, the location name of the current server is used.

The Compare Add Schema panel shows the schemas that match the partial schema name (R%) that was specified on the Compare Add Schema panel.

```

Compare ----- DSN8 Compare Add Schema ----- Row 1 of 3
Command ==>>                                     Scroll ==>> PAGE

Valid line commands are:                            Location: STPLEX4A_DSN8
S - Select (add)

      Schema
Select Name  Action
-----
      RAJESHR
      RMILLAR
      R148286
  
```

Figure 27. Compare Add Schema panel (GOC1CD)

The Compare Add Schema panel contains the following fields:

Schema Name

This column shows all of the schemas that match the partial name you specified in the DB2 Compare Add Schema process.

Action

This field shows the action that was performed for the corresponding schema.

Location

This field indicates the unique location name for an accessible server. If you did not specify a location name, the location name of the current server was used.

- Issue the Select command in the Select field that corresponds to the schema to add to the list of schemas to be compared or issue the ALL command to select all objects in the list.

Specifying the input version file data set name

Select the Specifying the input version file data set name option to specify the data set name of the version file that is to be used as input to the compare process (source or target).

About this task

To specify the input version file data set name:

- Select option 3 on the Specify Compare Source panel.

If Change Management is enabled on your system, the Specify Source Compare Version File panel, as shown in the following figure, is displayed. If Change Management is not enabled, the Version table Owner and Name fields are not available on this panel.

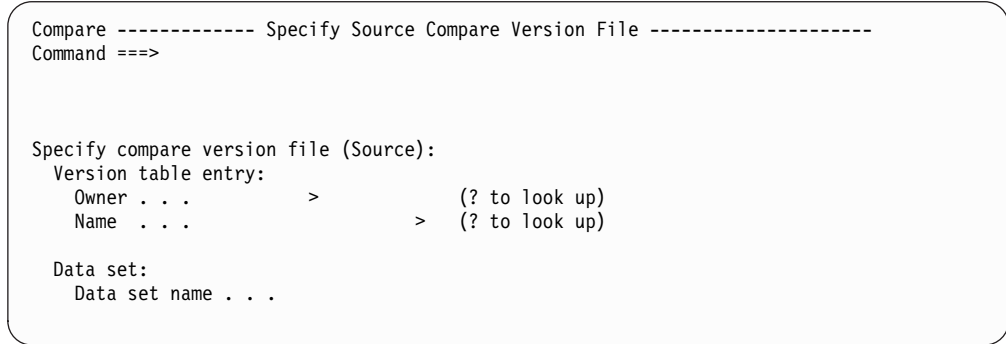


Figure 28. Specify Source Compare Version File panel (GOC13)

2. If you are using Change Management, specify the version table owner and name. Enter a question mark (?) in the field to display the CM Versions panel (ADB2C41) from which you can select a version file from the Change Management database.

Attention: If you select the option to store the version file in the database, an additional step is created in the compare job to store the version file for both the source and target objects in the database. If both the database and the data set are specified, the information in the data set field is used to determine the output destination for the version file.

Refer to “Managing changes to DB2 objects” in *IBM DB2 Administration Tool for z/OS User’s Guide and Reference* for additional information about Change Management.

3. Alternatively, if you are not using Change Management, specify the data set name of a previously created version file. The data set must be either:
 - A variable-block sequential data set
 - A member of a partitioned data set

The data set must have a record length of 16K (RECFM=Vx, LRECL=16384). The records in this data set must be the output from running an extract of either a DDL file or objects in the DB2 catalog.

4. Press Enter to process your input and return to the Object Comparison Tool Main Menu.

Related tasks:

“Specifying that the source is from an extract from the DB2 catalog” on page 67 Use the Specify DB2 Source Catalog Extract panel to specify the DB2 catalog extract option, a version file output data set, and an optional description of the objects that you are comparing.

Specifying that the source is a version scope

You can compare object types by defining a version scope for objects. A version scope can contain databases, table spaces, tables, indexes, views, stored procedures, triggers, and other objects.

About this task

Version scopes are stored in change management tables. You must be set up with Change Management to use the version scope option. Refer to “Prerequisites for Change Management” in *IBM DB2 Administration Tool for z/OS User’s Guide and Reference* for additional information about Change Management set up.

Use the following steps to specify that the source is from the DB2 catalog and that the objects are selected from a version scope.

1. Select option **1 - Specify compare source (new)** from the DB2 Object Comparison Tool Menu (GOCMENU).
2. On the Specify Compare Source panel (GOC1), enter VS to select objects from a version scope.

```
GOC1 re ----- Specify Compare Source ----- 17:01
Option ==> VS

1 - Source is from a DDL file
2 - Source is from the DB2 catalog
3 - Source is from a compare version file

VS - Source is from the DB2 catalog and the objects are selected from
a version scope
```

Figure 29. Specify Compare Source panel (GOC1)

3. Since you are using Change Management, you must specify the version scope owner and name. On the Specify Source Version Scope panel (GOC1VS), enter ? to look up the name of the version scope that you want to compare.

```
GOC1VS ----- Specify Source Version Scope -----
Command ==>

Specify version scope (Source):
Version Scope entry:
Owner . . . SMITHAJ > (? to look up)
Name . . . ? > (? to look up)
```

Figure 30. Specify Compare Source panel (GOC1VS)

The CM - Version Scopes panel (ADB2C42) displays the names of version scopes that are available.

Refer to “Version scopes” in *IBM DB2 Administration Tool for z/OS User’s Guide and Reference* for additional information about version scopes.

```
ADB2C42 n ----- CM - Version Scopes ----- Row 1 to 4 of 4
Command ==> Scroll ==> PAGE
Select by typing '+'
Line commands:
VE - Versions SO - Version scope objects GV - Generate new base version
INS - Insert U - Update DEL - Delete I - Details on version scope

Sel          ID Owner      Name                Comment
          * *          *                *
-----
          601 SMITHAJ  D7348
          581 SMITHAJ  SCOPETEST
          501 SMITHAJ  SCOPE1
+          461 SMITHAJ  SCP1
***** END OF DB2 DATA *****
```

Figure 31. CM - Version Scopes (ADB2C42)

4. On the Specify Source Version Scope panel, enter the name of the version scope source that you want to compare.

```

GOC1VS ----- Specify Source Version Scope -----
Command ==>

Specify version scope (Source):
Version Scope entry:
  Owner . . . SMITHAJ >          (? to look up)
  Name . . . SCOPE1           >  (? to look up)

```

Figure 32. Specify Source Version Scope panel (GOC1VS)

5. Return to the GOCMENU panel. The status is now shown for **Version scope specified (DB2 catalog)**.

```

GOCMENU ----- DB2 Object Comparison Tool Menu ----- 17:13
Option ==> 2

          Status:
1 - Specify compare source (new)      Version scope specified (DB2 catalog)
2 - Specify compare target (old)     Incomplete
3 - Specify compare masks             None specified
4 - Specify fields to ignore          Using defaults
5 - Generate compare job              Not generated

W - Walk through steps 1 - 5 in sequence
V - Generate job to extract version file from source only

R - Reset all
RS - Reset source
RT - Reset target

S - Save dialog
M - Manage/Restore dialog
MC - MultiCompare

```

Figure 33. DB2 Object Comparison Tool Menu (GOCMENU)

Related concepts:

Chapter 7, “Batch compare program,” on page 157

The batch compare program is run when you specify options on the Generate Compare Jobs panel and generate a compare batch job. This program compares two sets of DB2 objects, reports all differences, and writes all changes to a file. This file is used to generate updates to upgrade target objects to the level of source objects.

Related tasks:

“Generating a compare batch job” on page 105

Select the Generate compare job option to specify the parameters for generating the batch compare job.

Exclude objects from the compare process

You exclude objects from the compare process to be more selective about what object types participate or do not participate in the compare process of DB2 Object Comparison Tool.

The exclude process

Use the Exclude Specification panel (ADBPC71) before you compare objects to create a list of objects that DB2 Object Comparison Tool will exclude from the

source, target, or both. Specify the objects to be excluded by issuing the ESL line command and proceeding to the Exclude Objects panel (ADBPC7L).

Restriction: You cannot use the Exclude Specification panel when you are analyzing or running a change in Change Management.

You can set the duration for use of an exclude specification. After the date is passed, the exclude specification is eligible for deletion. You can change an auto-delete date by modifying the exclude specification.

The exclude specification on an object does not cascade to its dependent objects. You must list all objects to exclude. For example, the exclusion of a table does not mean that its dependent objects, such as index and view, are also excluded. Any objects to be excluded must be included in an exclude specification, except for history tables. If a temporal table is excluded, then its history table is also excluded. If the history table is specified in exclude specification, then both the temporal and history table are excluded. You can use the wildcard character (*) when you are specifying object names.

You can edit or delete an exclude object. You can insert or repeat multiple rows by issuing the respective line command followed by the number of rows (up to 99) that you want to insert or repeat. For example, I4 will insert 4 new rows and R6 will repeat the selected row 6 times.

When you specify objects for exclusion, you can also specify authorizations to exclude. You can exclude authorizations independent from the object they are associated with. Authorizations that you select for exclusion are entered into the exclude specification. When you use the DB2 Admin GEN function, the authorizations are excluded from DDL generation.

DB2 Object Comparison Tool does not add source authorizations to the target if the source authorization does not exist at the target. Instead, a warning is issued. If you exclude authorizations at the source, then no message is issued.

When comparing objects, if an object appears in the target only, the action is to drop the object. However, if the object is excluded, then it is not processed and the object is not dropped at the target. The object is retained. The same action occurs with authorizations. During the compare process, when you exclude an authorization, the authorization that already exists at the target is retained.

An implicit drop occurs when the action of dropping an object results in the drop of a dependent object. For example, you might want to be able to drop a table space, and as a result, also drop the table, index, and other objects dependent on the table space. On the Generate Compare Jobs panel (GOC5), you specify if objects can be dropped implicitly. If the option **Allow implicit drop of excluded objects** is specified as yes, then excluded objects can be dropped as needed and are re-created according to the target definition. If the option is NO, then if an excluded object needs to be dropped (perhaps because its parent object was dropped or because its dependent object was dropped) Object Compare stops processing compare and severe message is issued.

Objects you can exclude

To exclude an object, enter the two character code of the object type in the Exclude Specification panel. You can exclude objects for the following types:

- AL - Alias

- DB - Database
- DT - Distinct type
- FU - User-defined function
- GV - Global variable
- IX - Index
- RL - Referential constraint
- SC - Schema
- SG - Storage group
- SP - Stored procedure
- SQ - Sequence
- SY - Synonym
- TB - Table
- TG - Trigger
- TS - Table space
- VW - View

If an object type is schema, all UDFs, UDTs, Procedures, Triggers, and Sequences that are part of the schema are excluded. You can also exclude archive tables and archive-enabled tables. When an archive-enabled table is excluded, its corresponding archive table is also excluded, and vice versa. Similarly, when a temporal table is excluded, its history table is also excluded, and vice versa.

To exclude an authorization, enter the two character code of the authorization type in the Exclude Specification panel. You can exclude authorizations for the following types:

- CA - Column authorizations
- TA - Table authorization
- VA - View authorization

Creating or managing exclude specifications through Change Management

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 34. Manage Exclude Specifications panel (ADBPC7)

3. Select an option to view an existing specification or create a new specification.

| Option | Description |
|--|---|
| <p>Edit an existing exclude specification</p> | <ol style="list-style-type: none"> Specify Owner name or specification name. You can enter ? to look up a name from a list. Select Option 1 - Display exclude specification. 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. Exit and return to the CM - Manage Exclude Specifications (ADBPC7) panel. |
| <p>Create a new exclude specification</p> | <ol style="list-style-type: none"> Select Option 2 - Create an exclude specification. In the Create Exclude Specifications (ADBPC22) panel, you specify owner name and specification name. You also can specify an Eligible for auto-delete value. Press Enter and in the CM - Exclude Objects (ADBPC7L) panel, insert lines and enter object names and other information. Exit and return to the CM - Manage Exclude Specifications (ADBPC7) panel. |

Creating exclude specifications with an initial compare

You create exclude specifications to omit objects from the compare process. You use DB2 Admin Tools to specify objects that you want to exclude from the compare process.

About this task

Use the Exclude Specification panel (ADBPC71) to specify objects to exclude from sources and targets in the DB2 Object Comparison Tool. Excluded objects are treated as though they are not in the source or target. You might want to exclude objects so that DB2 Object Comparison Tool does not adjust processing based specified object.

The following task steps describe how to create exclude specifications from DB2 Object Comparison Tool. You can also create exclude specifications from Change Management.

Procedure

1. From the DB2 Admin Main Menu, specify option c. The DB2 Object Comparison tool (GOCMENU) panel is displayed.
2. Select the option to Specify Compare Source or Specify Compare Target and select how you want to create a new exclude specification.

| Option | Description |
|---|--|
| Edit an existing exclude specification | <ol style="list-style-type: none">1. Specify Owner name or specification name. You can enter ? to look up a name from a list.2. In the Edit objects field, specify YES.3. Press Enter and in the CM - Exclude Specifications (ADBPC71) panel, enter the ESL line command next to a listed specification.4. In the CM - Exclude Objects (ADBPC7L) panel, edit, insert, delete, or repeat lines and modify the object names and other information.5. Exit and return to the Specify Compare Source or Specify Compare Target panel. |
| Create a new exclude specification | <ol style="list-style-type: none">1. Type in an owner name and specification name.2. In the Edit objects field, specify YES.3. Press Enter and in the CM - Exclude Objects (ADBPC7L) panel, insert lines, and type in object names and other information.4. Exit and return to the Specify Compare Source or Specify Compare Target panel. |

Creating exclude specifications from stored compare results

You can create exclude specifications by using the saved compare results. You can select objects that you want to include in the exclude specification and exclude from the compare process.

About this task

Your saved compare results might include objects that you do not want to include in future compares. Use the DB2 Object Comparison Tool to find objects. You select

objects to exclude from sources, targets or both, though use of the CM - Compare Results (ADBPCRS) panel.

Procedure

1. From the DB2 Admin Main Menu, specify option c. The DB2 Object Comparison tool (GOCMENU) panel is displayed.
2. Select the option MR - Managed saved compare results.
3. In the Manage Compare Results (ADBPCR) panel, select the saved compare result that you want to work in. The Compare Results (ADBPCRS) panel is displayed.
4. From the Compare Results (ADBPCRS) panel, type one of the following line commands in the Select column, next to a compare result.

| Option | Description |
|---|--|
| EX - Exclude line command | The Create Exclude Specification (ADBPCEX) panel is displayed. You specify owner and name for the exclude specification. |
| EXS - Exclude from source line command | The Exclude Option (ADBPCL) panel is displayed. You see a list of source objects that are generated from the compare run. You can insert, delete, or repeat lines. You can enter object names and other information. |
| EXT - Exclude from target line command | The Exclude Option (ADBPCL) panel is displayed. You see a list of target objects that are generated from the compare run. You can insert, delete, or repeat lines. You can enter object names and other information. |

```

ADBPCEX n ----- Create Exclude Specification ----- 14:53
Command ==>>

Source Exclude Specification:
Owner . . . . . JSMITH > (Optional, default is JSMITH, ? to lookup)
Name . . . . . SRCIESPEC01 > (Required, ? to lookup)
Comment . . . . . >
Eligible for auto-delete . . . 30 (no of days, blank for no auto-delete)

Target Exclude Specification:
Owner . . . . . JSMITH > (Optional, default is JSMITH, ? to lookup)
Name . . . . . TGTIESPEC01 > (Required, ? to lookup)
Comment . . . . . >
Eligible for auto-delete . . . 30 (number of days, blank for no auto-delete)

```

Figure 35. Create Exclude Specification panel (ADBPCEX)

```

ADBPC7L n ----- Exclude Objects ----- Row 1 to 3 of 3
Command ==>                                     Scroll ==> CSR

Exclude specification lines for "JSMITH"."SRCIESPEC01"
Commands: CANCEL
Line commands:
  D - Delete  E - Edit  I - Insert  R - Repeat

Sel T Qual      Name                Column/      Auth
    * *         *                   Constrnt  Grantee    Level      Message
----->----->----->----->----->----->----->----->----->
TB VNDRG   TB23367
SA DB23367 TS23367B
SA DB23367 TS23367B

```

Figure 36. Exclude Option panel (ADBPC7L)

Specifying target objects to be compared

Use the Specify Compare Target option to specify the DB2 target objects to be compared.

About this task

Use the following steps to specify the target object definitions:

1. Select option 2 from the DB2 Object Comparison Tool Menu to display the Specify Compare Target panel:

```

GOC1 re ----- Specify Compare Target ----- 17:15
Option ==>

1 - Target is from a DDL file
2 - Target is from the DB2 catalog
3 - Target is from a compare version file
4 - Target is from the DB2 catalog and the objects are automatically
   selected based on the selected source objects
VS - Target is from the DB2 catalog and the objects are selected from
    a version scope

Exclude Specification:
Owner . . . . . > (Optional, default is ELACZ, ? to lookup)
Name . . . . . > (Required, ? to lookup)
Edit objects . . . . YES (Yes/No)

```

Figure 37. Specify Compare Target panel (GOC1)

2. Specify the target for the DB2 objects that will be used in the compare job that you are creating:
 - Select option 1 to specify that the definition of the target DB2 objects is from a DDL file, that is, a file that contains SQL CREATE statements.
 - Select option 2 to specify that the definition of the target DB2 objects is from an extract from the DB2 catalog, that is, an extract of one or more databases, table spaces, or tables and all the dependent objects.
 - Select option 3 to specify that the definition of the target DB2 objects is from a previously created version file.
 - Select option 4 to specify that the target DB2 objects are from the DB2 objects that were selected from the source. DB2 Object Comparison Tool uses the

|
|
|

source object names, in combination with masks and renames that are to be applied, to determine the target object names. DB2 Object Comparison Tool then extracts the target objects accordingly.

- Select option VS to specify that the target is from the DB2 catalog and that the objects are selected from a version scope.

The process for specifying target objects in options 1 through 3 and option VS is the same as the process for specifying source objects.

If you select option 4, the Specify DB2 Location panel is displayed:

```
Compare ----- Specify DB2 Location -----  
Option ==>  
  
Specify location name: DB8A  
  
Specify output compare version file:  
Version data set : VERLIB.NEW(V2)  
  
Enter a description (optional):  
Description . . : APPLICATION V2  
  
----- DB2 Object Compare Warning -----  
Target objects will be automatically selected based on the objects you  
selected for the source. If the source version file does not include all  
dependent objects, any subsequent comparison with the target may lead to  
a loss of these dependent objects. To prevent a loss of objects, select  
"Suppress DROP of objects" when you generate the compare job(s).  
Press ENTER to continue or END to stop this operation.  
Please read this carefully
```

Figure 38. Specify DB2 Location panel (GOC14)

For information about preventing loss of dependent objects and the Suppress DROP of objects option, refer to “Generating a compare batch job” on page 105.

Procedure

1. Specify the DB2 target location name.
2. Specify the name of the output data set for the version file. DB2 Object Comparison Tool determines the set of objects that are defined in the source version file, and ADB2GEN extracts the definitions of these objects from the DB2 catalog. If an existing data set is specified, it is overwritten. The specified data set must be one of the following types:
 - A variable-block sequential data set
 - A member of a partitioned data set with a record length of 16K (RECFM=Vx, LRECL=16384)

If the specified version data set does not exist, it is created.

3. Optionally, in the description field, specify a description of the DB2 objects that you are comparing. The description is printed in the Comparison report, is placed in the header record of the version file, and is used to describe the common properties of the DB2 objects. Example: APPLICATION V2

Results

Attention: DB2 Object Comparison Tool ignores different buffer pool names if one of the names is derived from DDL text that does not provide a specific buffer pool name. For example, if the source that was specified with a DDL text contains a CREATE INDEX statement that does not contain a buffer pool clause, the buffer pool names are not compared. DB2 Object Comparison Tool issues a warning when buffer pool names are not compared because a buffer pool clause was not specified.

Related tasks:

“Specifying source object definitions to be compared” on page 64

The first step in defining the attributes of the DB2 objects to compare is to specify if the source object is from a DDL file, from the DB2 catalog, or from a compare version file. Use the Specify Compare Source panel to define these attributes.

Specifying the compare masks data set

Use the Specify Compare Masks panel to specify the name of the data set that contains the translation masks. You can also use this panel to specify whether you want to edit that data set now.

About this task

Prerequisite: Before you specify a compare masks data set, you must first have created the data set and populated it with mask values.

To specify the compare masks data set:

1. Select option 3 on the DB2 Object Comparison Tool Menu to display the Specify Compare Masks panel. If Change Management is enabled on your system, the Specify Compare Masks panel, as shown in the following figure, is displayed. If Change Management is not enabled, only the Mask data set and Edit Masks fields appear on the panel.

```
Compare ----- Specify Compare Masks -----
Option ==>

Mask Table Entry:
  Owner . . MASK > (? to look up)
  Name . . MASKNAME > (? to look up)
Data Set:
  Mask DSN . .
Options:
  Edit Mask . . YES (Yes/No)
```

Figure 39. Specify Compare Masks panel (GOC3)

Refer to “Managing changes to DB2 objects” in *IBM DB2 Administration Tool for z/OS User’s Guide and Reference* for additional information about using masks and the Specify Compare Masks panel when Change Management is enabled.

2. If you are using Change Management, specify the version table owner and name. Use the question mark (?) to look up an existing version file.

Attention: If both the mask table entry and the data set are specified, the information in the data set field is used to determine the output destination for the version file.

- Specify if you want to edit the mask in the Edit Mask field and press Enter. If you specify No, the Object Comparison Tool main menu is displayed. If you specify an existing mask, the Mask Lines panel is displayed. If you specify Yes and have created a new mask to store in the Change Management database, the following figure is displayed:

```

DB2 Admin ----- CM - Insert Mask ----- 12:25
Command ==>

Owner . . . MYID    > (Optional, default is USERID)
Name . . . MYMASK  > (Required, ? to look up)
Comment . . .                                     >

```

Figure 40. Insert Mask panel (ADB2C22)

- Optionally, enter a comment in the comment field.
- Press Enter to execute the INSERT statement. A confirmation message is displayed.
- Press F3 to go to the Masks Lines panel, shown in the following figure:

```

DB2 Admin ----- CM - Mask Lines ----- Row 1 to 1 of 1
Command ==>                                     Scroll ==> PAGE

Mask lines for mask "MASK"."MASKNAME"
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 41. Mask Lines panel (ADB2CL2)

The Mask Lines panel is used to define the mask.

- Define the lines to include in the mask and issue the SAVE command. The INSERT statement executed confirmation message is displayed.
- Press F3 to return to Object Comparison Tool main menu.
- Alternatively, if you are not using Change Management, in the Mask data set field, specify the name of the data set that contains the translation masks that are to be used for the compare process in the Mask data set name field. 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)
- In the Edit field indicate whether you want to edit the mask data set by using ISPF edit.

The Edit Compare Masks panel is displayed when you specify Yes in the Edit data set field of the Specify Compare 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,TBNAME,IXNAME,UDFNAME,CONSNAM
==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>   BPNAM
==MSG>   TSBPNAM,IXBPNAM
==MSG>   SGNAME
==MSG>   TSSGNAM,IXSGNAM
==MSG> AUTHID
==MSG> SQLID
==MSG> SCHEMA
==MSG>   IXSCHEMA,PMSHEMA,MKSCHEMA,SETPATHSC
==MSG>   TGSHEMA,UDTSHEMA,SEQSCHEMA,STPSCHEMA
==MSG>   UDFSHEMA,GVSCHEMA
==MSG>   TBSHEMA
==MSG>   ALSHEMA,VWSCHEMA
==MSG> OWNER
==MSG>   DOWNER,TOWNER,IXOWNER,SGOWNER,PKGOWNER
==MSG>   TOWNER
==MSG>   SYOWNER
==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 (for tables)
==MSG> DTINLOBL n,REXX exit (for distinct types)
==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)

```

Figure 42. Edit Compare Masks panel (GOEDIT), 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>   TBSHEMA 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. VVNAME 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 TBSHEMA
==MSG>   for schema; therefore, it is recommended to use both VVNAME
==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> Object-specific mask examples:
==MSG>   TBSHEMA:CREATOR1.TB2:CREATOR1,NEW_CRE1
==MSG>   IXNAME:IXOWN*.IX3*:IX3*,IX4*
==MSG>   IXBPNAME:IXOWN1.INDX2:BP1,BP3
==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> Syntax for info about renamed objects/columns:
==MSG>   renameobj:old-name,new-name
==MSG>   RENAMCOL: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>   RENAMCOL:OWNER.MYTB.OLDCOLNAME,NEWCOLNAME
***** Bottom of Data *****

```

The message lines on the panel list the available translation mask names.

Notice that there is a hierarchy on the panel. For example, to change a DBNAME, specify the following mask:

```
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 bufferpool 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.

Related concepts:

“Translation masks” on page 187

Use translation masks to allow a match to be found when the compare source and target objects use different naming conventions. If you specify both mask and ignore, however, the ignore overrides the mask.

Related reference:

Chapter 12, “DB2 catalog records and associated masks,” on page 201

The Mask application details table shows the fields in DB2 catalog records that have masks applied before the compare process takes place or before GEN creates DDL.

Specifying compare ignore fields

Specify compare ignore fields when you want to ignore some fields when comparing DB2 catalog records. Use ignore fields when there are differences in source and target objects, but you don't want the compare process to change them.

About this task

Use the following steps to specify the name of the data set that contains the ignore fields and to indicate whether you want to modify that data set:

1. Select option 4 on the DB2 Object Comparison Tool Menu to display the Specify Compare Ignore Fields panel. If Change Management is enabled on your system, the Specify Compare Ignore Fields panel, as shown in the following figure, is displayed. If Change Management is not enabled, only the Ignore data set and Edit Ignores fields appear on the panel.

```

Compare ----- Specify Compare Ignores -----
Option ==>

Ignore Fields Specification:
Owner . . . . . > (? to look up)
Name . . . . . > (? to look up)
Data Set:
  Data Set Name . .
Options:
  Edit Ignore Fields Specification . . . NO (Yes/No)

Ignore Changes Specification:
Owner . . . . . > (? to look up)
Name . . . . . > (? to look up)
Edit Ignore Changes Specification . . . NO (Yes/No)
Display using a saved compare result . . NO (Yes/No)
  Saved Compare Results:
    Owner . . . . . > (? to look up)
    Name . . . . . > (? to look up)

```

Figure 44. Specify Compare Ignores panel (GOC4)

Refer to “Managing changes to DB2 objects” in *IBM DB2 Administration Tool for z/OS User’s Guide and Reference* for additional information about using ignore fields and the Specify Compare Ignore Fields panel when Change Management is enabled.

You do not need to complete the Ignore Changes Specification section of the panel.

2. If you are using Change Management, specify the ignore table owner and name. Use the question mark (?) to look up an existing version file.

Attention: If both the ignore table entry and the data set are specified, the information in the data set field is used to determine the output destination for the version file.
3. Specify if you want to edit the ignore fields in the Edit Ignores field and press Enter. If you specify No, the Object Comparison Tool main menu is displayed. If you specify an existing ignore field and Yes to edit, the Specify Ignore Fields: Objects panel is displayed. If you specify Yes and have created a new ignore field to store in the Change Management database, the following figure is displayed:

```

DB2 Admin ----- Create Ignore Specification ----- 12:25
Command ==>

Owner . . . MYID > (Optional, default is USERID)
Name . . . MYIGNORE > (Required, ? to look up)
Comment . . . . . >

```

Figure 45. Insert Ignore panel (ADB2C22)

4. Optionally, enter a comment in the comment field.
5. Press Enter to execute the INSERT statement. A confirmation message is displayed.
6. Press Enter again to display the Specify Ignores Field: Objects panel.
7. Alternatively, if you are not using Change Management, specify the name of the data set that contains the ignore fields to be used during the compare

process. The data set must contain ignore fields, 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 logical record length of 80 (RECFM=Fx, LRECL=80)

The input must be in columns 1-72 of the ignore data set.

- Specify whether you want to modify (add or delete fields) the data set and press Enter.

If you specify Yes, the Specify Ignore Fields: Objects panel is displayed:

```

Compare ----- Specify Ignore Fields: Objects ----- Row 1 of 17
Command ==>                                           Scroll ==> PAGE

Valid line commands are:
U - Update Ignore Fields

Select Object          Ignore Fields
*                    *
-----
GENERIC               None
SYSCHECKS             None
SYSCOLUMNS          COLTYPE,LENGTH,SCALE,DEFAULT,DEFAULTVALUE
SYSDATABASE           CREATOR,STGROUP,BPOOL,INDEXBP
SYSDATATYPES         None
SYSFIELDS             None
SYSINDEXES           None
SYSINDEXPART         PQTY,SQTY,SECQTYI
SYSKEYS               None
SYSPARMS             None
SYSRELS              None
SYSROUTINES          None
SYSSEQUENCES         None
SYSSTOGROUP          VCATNAME
SYSTABLEPART         PQTY,SQTY,SECQTYI
SYSTABLES            None
SYSTABLESPACE        None
SYSTRIGGERS          None
SYSVIEWS             None
SYSVOLUMES           VOLID
  
```

Figure 46. Specify Ignore Fields: Objects panel (GOCCI)

The Specify Ignore Fields: Objects panel shows the objects and the fields within the objects, if any, that are to be ignored. Several objects have ignore fields defined. For example, for the object SYSDATABASE, the fields CREATOR, STGROUP, BPOOL, and INDEXBP are to be ignored.

The Specify Ignore Fields: Objects panel contains the following fields:

Object

This field contains the names of the DB2 catalog tables that describe (part of) the object.

Ignore Fields

This field contains the fields to ignore for this DB2 catalog table.

- Issue the U (update) command to modify the ignore fields. The Select Ignore Fields panel for the selected object, is displayed:

```

Compare ----- Select Ignore Fields for SYSDATABASE Row 1 to 11 of 11
Command ==>                                     Scroll ==> PAGE

Valid line commands are:
S - Select (add) field U - Un-select

Select Fields          Action
*                      *
-----
*    CREATOR           Selected
*    STGROUP           Selected
*    BPOOL             Selected
    ROSHARE
    TYPE
    GROUP_MEMBER
    ENCODING_SCHEME
    SBCS_CCSID
    DBCS_CCSID
    MIXED_CCSID
*    INDEXBP          Selected
***** END OF DB2 DATA *****

```

Figure 47. Select Ignore Fields panel (GOCCIF)

- 10. Select the fields that are to be ignored and press Enter.
- 11. Press PF3 to return to the DB2 Object Comparison Tool Menu.

Related concepts:

“Ignore fields” on page 194
 By using ignore fields, you can compare DB2 catalog records while ignoring some fields. Ignore fields are used in situations where you are aware of differences between source and target objects, but you do not want these changes to be recognized and cause a change.

Selecting SYSINDEXPART ignore fields

The Select Ignore Fields for SYSINDEXPART panel shows the ignore fields that you specified for the SYSINDEXPART object on the Specify Ignore Fields: Objects panel.

About this task

```

GOCCIF ----- Select Ignore Fields for SYSINDEXPART ---- Row 1 of 11

Valid line commands are:
S - Select (add) field U - Unselect

Select Fields          Action
*                      *
-----
    PQTY              Selected
    SQT              Selected
    STORTYPE
    STORNAME
    VCATNAME
    FREEPAGE
    PCTFREE
    INDEXTYPE
    GBPCACHE
    SECQTYI          Selected

```

Figure 48. Select Ignore Fields for SYSINDEXPART panel (GOCCIF)

Issue the Select command in the Select field that corresponds to the field to add to the list of ignore fields. Use the Unselect line command to remove a field from the list.

Fields These are the DB2 catalog table fields that you can set to be ignored for the SYSINDEXPART object you selected in the Specify Ignore Fields: Objects panel. In this case, the three fields that were specified for SYSINDEXPART (PQTY, SPTY, and SECQTYI) are set to be ignored during the compare process.

Action

This field shows the action that was performed for the corresponding field.

Press PF3 to return to the Specify Ignore Fields: Objects panel.

Selecting SYSDATABASE ignore fields

The Select Ignore Fields for SYSDATABASE panel shows the ignore fields currently specified for the SYSDATABASE object on the Specify Ignore Fields: Objects panel.

About this task

```
GOCCIF ----- Select Ignore Fields for SYSDATABASE ----- Row 1 of 11

Line commands:
S - Select (add) field U - Un-select

Select Fields          Action
*                      *
-----
CREATOR                Selected
STGROUP                Selected
BPOOL                  Selected
ROSHARE
TYPE
GROUP_MEMBER
ENCODING_SCHEME
SBCS_CCSID
DBCS_CCSID
MIXED_CCSID
INDEXBP                Selected
```

Figure 49. Select Ignore Fields for SYSDATABASE panel (GOCCIF)

Issue the Select command in the Select field that corresponds to the field to add to the list of ignore fields. Use the Unselect line command to remove a field from the list.

Fields These are the DB2 catalog table fields that you can set to be ignored for the SYSDATABASE object you selected in Figure 46 on page 97. In this case, the four fields that were specified for SYSDATABASE (CREATOR, STGROUP, BPOOL, and INDEXBP) are set to be ignored during the compare process.

Action

This field shows the action that was performed for the corresponding field.

Press PF3 to return to the Specify Ignore Fields: Objects panel.

Selecting SYSCOLUMNS ignore fields

The Select Ignore Fields for SYSCOLUMNS panel shows the ignore fields currently specified for the SYSCOLUMNS object on the Specify Ignore Fields: Objects panel.

About this task

Caution: Special care should be taken when you 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.

When specifying ignore fields for SYSCOLUMNS, you should 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 DEFAULT can cause the SYSSEQUENCES row to be included or excluded, depending on the value of DEFAULT 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 FOREIGNKEY 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 FLDPROC 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.

```
GOCCIF ----- Select Ignore Fields for SYSCOLUMNS Row 1 to 14 of 14

Valid line commands are:
 S - Select (add) field U - Un-select

Select Fields          Action
*                      *
-----
      COLTYPE          Selected
      LENGTH           Selected
      SCALE            Selected
      NULLS
      REMARKS
      DEFAULT          Selected
      KEYSEQ
      FOREIGNKEY
      FLDPROC
      LABEL
      DEFAULTVALUE     Selected
      LENGTH2
      TYPESCHEMA
      TYPENAME
```

Figure 50. Select Ignore Fields for SYSCOLUMNS panel (GOCCIF)

Fields These are the DB2 catalog table fields that you can set to be ignored for the SYSCOLUMNS object that you selected in Figure 46 on page 97. In this case, the fields that were specified for SYSCOLUMNS (COLTYPE, LENGTH, SCALE, DEFAULT, and DEFAULTVALUE) are set to be ignored during the compare process.

Action

This field shows the action that was performed for the corresponding field.

Press PF3 to return to the Specify Ignore Fields: Objects panel.

Creating ignore changes specifications

You can create ignore changes specifications by using the saved compare results. You can select objects changes that you want to include in the ignore changes specification and ignore during the compare process.

About this task

Your saved compare results might include objects changes that you do not want. Object changes that you specify as ignore are reported but no SQL statement is generated for the change. Only changes that report difference between source object and target object can be ignored. Added and dropped objects can be excluded from compare processing but not ignored.

When you select object changes to be ignored be careful. Many objects and fields in the DB2 catalog records are interdependent. When one change is ignored, another change might be invalid if it is not also ignored. For example, if change to the number of table space partitions is ignored, then other object changes related to partitioning need to be ignored. In this case, changes to the number of partitions in the table, and adding or deleting limit key must also be ignored. LOB columns and LOB objects are another example of interdependencies. If you add a LOB column to a table is ignored, then adding an explicit auxiliary table for the column must also be ignored and its explicit LOB table space and index be excluded from compare. You must specify each object change that you want ignored. Related object changes are not automatically ignored.

Ignore changes for tables, global variables, indexes, and user-defined distinct types (UDT) are supported. Temporal and history tables, materialized query tables (MQT), and hash organization changes cannot be selected to be ignored.

Procedure

1. From the DB2 Admin Main Menu, specify option C. The DB2 Object Comparison tool (GOCMENU) panel is displayed.
2. Select the option MR - Managed saved compare results.
3. In the Manage Saved Compare Results (ADBPMCR) panel, select the saved compare result that you want to work in. The Manage Saved Compare Results (ADBPCR) panel is displayed.
4. In the Manage Saved Compare Results (ADBPCR) panel, type the RPT command. The Compare Report (ADBPCRR) panel is displayed.
5. In the Compare Report (ADBPCRR) panel, type the I - Ignore change line command in the Select column, next to compare changes that you want ignored in subsequent change processing.

```

ADBPCRR ----- DSNAP Compare Report ----- 08:22
Command ==> Scroll ==> PAGE

Compare analysis report for "OWN1"."NEW1"
Commands: CONTINUE IGNOREALL RESETALL
Line commands:
  I - Ignore change R - Reset

Sel S Report

----->
Compare database source(DB23367) and target(DB23367)
  No changes to database

Compare tablespace source(DB23367.TS23367A) and target(DB23367.TS23367A)
  (A)Field BUFFERPOOL changed from BP0 to BP1
  Tablespace will be altered

Tablespace DB23367.TS23367B not found on target
  New Tablespace DB23367.TS23367B will be added
  Authorizations for Tablespace DB23367.TS23367B will be copied from source

Compare table source(VNDRG.TB23367) and target(VNDRG.TB23367)
  Column LAHMANID
I   (A)Type changed from VARCHAR(12) to VARCHAR(20)
   (D)Nulls/default changed from NOT NULL to WITH DEFAULT NULL
  Column DATE_YYYY
I   (A)Type changed from SMALLINT to INTEGER
   (D)Column TEAM_NAME added
  Table VNDRG.TB23367 will be dropped
  Table will be recreated
  Table data will not be converted
  Not eligible for FORMAT INTERNAL processing

Compare index source(VNDRG.IX23367) and target(VNDRG.IX23367)
  Index VNDRG.IX23367 will be dropped by dropping the table
  Index will be recreated because the base table will be dropped and recreated

```

Figure 51. Compare Report panel (ADBPCRR)

When you press the Enter key, the status column for an object type that is selected for ignore is updated to 'I'.

6. While still in the Compare Report (ADBPCRR) panel, type the CONTINUE command. The Create Ignore Specification (ADB2C22) panel is displayed.
7. In the Create Ignore Specification (ADB2C22) panel, type in an owner name and specification name. You now have a specification that you can reference when you run your compare process. 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.

Managing ignore changes specifications

You manage lists of object changes that are ignored during the compare process by maintaining ignore changes specifications.

About this task

Ignore changes for tables, global variables, indexes, and user-defined distinct types (UDT) are supported. Temporal and history tables, materialized query tables (MQT), and hash organization changes cannot be selected to be ignored.

Procedure

1. From the DB2 Admin Main Menu, specify option C. The DB2 Object Comparison tool (GOCMENU) panel is displayed.
2. Select option 4 - Specify ignores. The Specify Compare Ignores (GOC4) is displayed.

```

Compare ----- Specify Compare Ignores -----
Option ==>

Ignore Fields Specification:
Owner . . . . . > (? to look up)
Name . . . . . > (? to look up)
Data Set:
  Data Set Name . .
Options:
  Edit Ignore Fields Specification . . . NO (Yes/No)

Ignore Changes Specification:
Owner . . . . . OWN1 > (? to look up)
Name . . . . . NEW1 > (? to look up)
Edit Ignore Changes Specification . . . YES (Yes/No)
Display using a saved compare result . . NO (Yes/No)
Saved Compare Results:
  Owner . . . . . OWN1 > (? to look up)
  Name . . . . . ISPEC01 > (? to look up)
  
```

Figure 52. Specify Compare Ignores panel (GOC4)

3. In the **Edit Ignore Changes Specification** field, specify YES. The Ignored Changes List (ADBPCICL) is displayed.

```

ADBPCICL ----- Ignored Changes List ----- Row 1 to 14 of 30
Command ==> Scroll ==> PAGE

Ignored changes for "OWN1"."ICSPEC01"
Line commands:
D - Delete

Sel T Target Qual Target Name Attribute Additional Info
----->----->----->----->
      DB23367 TS23367A BUFFERPOOL
TB VNDRG TB23367 Data type LAHMANID
TB VNDRG TB23367 Data type DATE_YYYY
  
```

Figure 53. Ignored Changes List (ADBPCICL)

You can add object changes by using the wildcard character (*), or delete an object change that is listed. You add a new object type and name in the blank line that is located below the column headings and before the listed change objects. When you add an object change, you must specify the object type.

When you add an object change, all changes to the object are ignored. Using the wildcard character (*) in the Target Qualifier or Target Name column forces all changes for objects that match what you specify to be ignored. Ignored changes are not applied to the target objects. For example, if you specify new* to be ignored, then objects that met the wildcard specification new* are still processed. However, all changes to all objects that met the wildcard specification are ignored.

If the Target Qualifier or Target Name are left blank, an asterisk (*) is substituted. If the qualifier or name does not include a wildcard character, then the wildcard character (*) is appended to the qualifier or name.

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.

Managing ignore specifications through Change Management

You use DB2 Admin Tools to specify object changes 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
Altered within
Eligible for auto-delete:
Within . . . . .
Next . . . . .
```

Figure 54. 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

Sel Owner      Name                                     Eligible for
                                     auto-delete Comment
-----
OWN1      ICSPEC01                                     2012-12-31
OWN1      ICSPEC02
```

Figure 55. 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.

Generating a compare batch job

Select the Generate compare job option to specify the parameters for generating the batch compare job.

About this task

Prerequisite: You must specify a compare source and a compare target before you can generate a compare job.

Use the following steps to specify the options for generating a compare batch job:

1. Select option 5 on the DB2 Object Comparison Tool Menu to display the Generate Compare Jobs panel:

```

GOC5 ----- Generate Compare Jobs -----
Command ==>

Specify the following for DB2 Object Comparison Tool:

Worklist information:
  Worklist name . . . . . PQ76055N (also used as middle qualifier in DSNs)

Compare options:
  Suppress DROP of objects . No          (Yes/No)
  Suppress DROP of columns . No          (Yes/No)
  Suppress adding columns . No          (Yes/No)
  Run SQLID . . . . .                   (Blank, an SQLID, or <NONE>)
  Run Validate. . . . . V                (Validate, None)
  Object Grantor . . . . .               (Blank or an SQLID)
  Allow implicit DROP of
    excluded objects . . . . N          (Yes/No)
  Enable auth-switching . . . YES       (Yes/No)

Change reporting options . . Yes        (Yes/No)
Save compare results . . . . Yes       (Yes/No)

Data set information:
  PDS for batch jobs . . . . CMP.PQ76055N
  Prefix for data sets . . . NBRON
  Changes file data set name.
    Member name . . . . .                (if Changes file is an existing PDS)

Options:
  Generate online . . . . . Yes          (Yes/No)
  Single compare job . . . . No          (Yes/No)
    Member name . . . . . COMPARE       (default COMPARE)
  Generate apply jobs . . . . Yes        (Yes, No, or (Delta) Change)
    Generate one job. . . . . Yes       (Yes, No, or (Per) Process)
    Member prefix . . . . . APPLY       (default APPLY)
  As work statement list . Yes          (Yes/No to append to work stmt list)
  Use customized util opts. Yes         (Yes/No)
  Unload Method . . . . . : P           (Unload, Parallel unload, HPU)
  Generate templates. . . . No          (Yes/No)
  Stop on conversion error. NO          (Yes/No)
  Use DEFER YES . . . . . YES          (Yes/No)
  Allow rotate parts . . . . YES        (Yes/No)
  Retain GENERATED ALWAYS:
    For ROWID . . . . . YES            (Yes/No)
    For ROW CHANGE TIMESTAMP. YES      (Yes/No)
  Retain sequence:
    START and RESTART values. NO        (Yes/No)
  IDENTITY START value . . . ORIGINAL   (Original, Computed)
  Mask ignored fields . . . . NO        (Yes/No)

Optional jobs after Reload or Alter:
  Run CHECK DATA . . . . Yes           (Yes/No)
  Take an image copy . . R              (after: Reload/Alter/Both/None)
  Run REORG/REBUILD . . . M            (Mandatory, All relevant, None)
  Run RUNSTATS . . . . . R             (after: Reload/Alter/Both/None)
  Run REBIND . . . . . Yes             (Yes/No)

BP - Change batch job parameters
TU - Specify TEMPLATE usage
UO - Customize utility options
CO - Change options common to change functions

```

Figure 56. Generate Compare Jobs panel (GOC5)

2. Specify the values for each of the options.
3. When you are finished specifying values, press Enter to run the job.

Related tasks:

“Specifying that the source is a version scope” on page 80

You can compare object types by defining a version scope for objects. A version scope can contain databases, table spaces, tables, indexes, views, stored procedures, triggers, and other objects.

Related reference:

“Generate compare job options” on page 113

Specify the criteria that you want to use to generate the work statement list or JCL for your compare job.

Saved compare results

You can save summary of information about compare processing from DB2 Object Comparison Tool. You can use the saved compare results to analyze data about the compare or to increase the efficiency of subsequent compare runs.

The saved compare results contain information about objects that were part of a compare run. Information that is provided in the results includes differences that are detected, changes to occur, and how changes are to be implemented. When you generate a compare job, you can save the compare results.

Saving compare results

You can save the results of compare processing as part of generating a compare. You can specify duration of the saved results.

Procedure

1. From the DB2 Admin Main Menu, specify option c. The DB2 Object Comparison tool (GOCMENU) panel is displayed.
2. Select option 5 - Generate compare job.
3. On the Generate Compare Jobs (GOC5) panel, type YES next to **Save compare results** and complete specifying a compare job. When you exit the panel, the Save Compare Results (ADB2C22) panel is displayed.
4. Type in an owner name and specification name.

```
ADB2C22 n -----Save Compare Results ----- 08:22
Command ===>

Owner . . . OWN1      > (Optional, default is VNDRG, ? to lookup)
Name . . . NEW1      > (Required, ? to lookup)
Comment . . MY FIRST COMPARE RESULT                                >
Eligible for auto-delete . . 30 (number of days, blank for no auto-delete)
```

Figure 57. Save Compare Results panel (ADB2C22)

Managing saved compare results

You can view and modify characteristics of a saved compare results.

About this task

You can refine a search for saved compare results, by using search criteria fields that are provided on the Manage Saved Compare Results (ADBPMCR) panel.

Procedure

1. From the DB2 Admin Main Menu, specify option c. The DB2 Object Comparison tool (GOCMENU) panel is displayed.

2. Select the option MR - Managed saved compare results.
3. On the Manage Saved Compare Results (ADBPMCR) panel, specify the saved compare results that you want to view or modify.

```

ADBPMCR ----- Manage Saved Compare Results ----- 16:36
Option ==>

Compare results information:
Owner . . . . . >
Name . . . . . NEW* >

Enter additional selection criteria:
Created within . . . 2012 - 2013
Altered within . . .
Eligible for auto-delete:
Within . . .
Next . . .

```

Figure 58. Manage Saved Compare Results panel (ADBPMCR)

4. In the Manage Saved Compare Results (ADBPCR) panel, use line commands to view more detail, modify, or delete saved compare results.

```

ADBPCR ----- Manage Saved Compare Results -----Row 1 to 14 of 30
Command ==> Scroll ==> PAGE
Line commands:
U - Update DEL - Delete S - Show summary I - Details on results
RPT - Compare Report
Sel Owner Name Eligible for Comment
auto-delete
----->----->
S OWN1 NEW1 2012-12-31 My first compare result
OWN2 NEW2
DEL OWN3 NEW3
OWN4 NEW4
OWN5 NEW5
OWN6 NEW6
OWN7 NEW7

```

Figure 59. Manage Saved Compare Results panel (ADBPCR)

5. Optional: In the Manage Saved Compare Results (ADBPCR) panel, type the S - Show summary line command in the Select column, next to compare process for which you want to view results. The Compare Results (ADBPCRS) panel is displayed.

```

ADBPCRS ----- Compare Results ----- Row 1 to 14 of 30
Command ==>> Scroll ==> PAGE

Compare results for "OWN1"."NEW1"
Commands: RPT VERSION SRCEX TGTEX SHOWSOURCE AUTH
Line commands:
EX - Exclude EXS - Exclude from source EXT - Exclude from target
EXA - Exclude Authorizations EXC Exclude Constraints

Sel 0 Target Target Compare
     0 Schema Name Additional Info Action I
     * * * * *
----->-----
DB DB23367 DB23367 No changes ?
TS DB23367 TS23367A Altered
TS Added
TB VNRG TB23367 Dropped/created
IX VNRG IX23367 Dropped/createdY
RL VNRG TCHILD TGTFPARENT Dropped/created

```

Figure 60. Compare Results panel (ADBPCRS)

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.

Object

The type of object that was compared as part of compare run. The valid object types are:

- AL - Alias
- AR - Auxiliary table
- CL - Clone table
- DB - Database
- DT - User defined type
- FU - Function
- GV - Global variable
- IX - Index
- MK - Column mask
- PK - Rebind package
- PL - Rebind plan
- PM - Row permission
- RL - Referential constraint
- SC - Schema
- SG - Storage group
- SP - Stored procedure
- SQ - Sequence
- SY - Synonym
- TB - Table
- TG - Trigger
- TS - Table space
- VW - View

Target Schema

The target object owner.

Target Name

The target object name.

Additional Information

Referential constraint name if the change is updating a referential constraint.

Compare Action

The type of change being performed for the object.

Implicit

Indicates an implicit drop or implicit drop and recreate.

If you enter the command **SHOWSOURCE**, the source object associated with the target is displayed in the column **Additional Information** column.

If you enter the command **VERSION**, the following fields are available on the panel:

Version

Instead of a **Compare Action** column, you see a **Version** column. The Version column shows the version number of native stored procedures and PL/SQL functions.

Active version

Instead of an **I** column, you see an **A** column. The A column indicates an active version of specific native stored procedures and PL/SQL functions.

- Optional: In the Manage Saved Compare Results (ADBPMCR) panel, type the RPT - Show summary line command in the Select column, next to compare process for which you want to view a report. The Compare Report (ADBPCRR) panel is displayed.

```

ADBPCRR ----- Compare Report ----- 08:22
Command ==>                               Scroll ==> PAGE

Compare report for "OWN1"."NEW1"

  Compare database source(DB23367) and target(DB23367)
  No changes to database

  Compare tablespace source(DB23367.TS23367A) and target(DB23367.TS23367A)
  (A)Field BUFFERPOOL changed from BP0 to BP1
  Tablespace will be altered

  Tablespace DB23367.TS23367B not found on target
  New Tablespace DB23367.TS23367B will be added
  Authorizations for Tablespace DB23367.TS23367B will be copied from source
  Compare table source(VNDRG.TB23367) and target(VNDRG.TB23367)
  Column LAHMANID
  (A)Type changed from VARCHAR(12) to VARCHAR(20)
  (D)Nulls/default changed from NOT NULL to WITH DEFAULT NULL
  Column DATE_YYYY
  A)Type changed from SMALLINT to INTEGER
  (D)Column TEAM_NAME added
  Table VNDRG.TB23367 will be dropped
  Table will be recreated
  Table data will not be converted
  Not eligible for FORMAT INTERNAL processing

  Compare index source(VNDRG.IX23367) and target(VNDRG.IX23367)
  Index VNDRG.IX23367 will be dropped by dropping the table
  Index will be recreated because the base table will be dropped and recreated

```

Figure 61. Compare Report panel (ADBPCRR)

Making compare changes through Change Management

Registering changes in Change Management simplifies the process of recording and tracking the changes that you make to your DB2 objects.

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*.

You can register the change on multiple target locations. In addition, you can specify an optional target profile as you register the multi-target change.

Procedure

1. Specify the options for your comparison as you normally would by using the following steps:
 - a. Specify the source objects to be compared
 - b. Specify the target objects to be compared
 - c. Specify compare masks
 - d. Specify fields to ignore
 - e. Specify options for the compare job, ensuring that you specify Yes to have Apply jobs generated

If Change Management is mandatory for your SQL ID, the CM - Register Options panel is displayed.

If Change Management is optional for your SQL ID, the following panel is displayed:

```
Compare ----- Generate Compare Jobs -----
Option ==>

Specify the following for DB2 Object Comparison Tool:

Worklist information:
Worklist name . . . . . PQ76055N (also used as middle qualifier in DSNs)
Co. -----
| DB2 Admin ----- DSN8 Change Management Prompt ----- 09:17 |
| Change Management is optional for SQLID: VNDRJP |
| Ch Do you wish to use Change Management for this function: (Yes/No) |
| Da |
|-----|
```

Figure 62. Change Management prompt (ADB2CMRO)

2. Specify Yes to process the compare change through Change Management. If you specify No, the job is generated with the Apply step and the changes are not registered in the Change Management database. The CM - Register Options panel is displayed:

```

ADB2CRO n ----- CM - Register Options ----- 11:27
Command ==>

Commands: CONTINUE                                DB2 System: DB2X
                                                DB2 SQL ID: J148286

Specify the following values to register a change:

Owner . . . . . J148286                > (Optional, Default is J148286)
Name . . . . . change1                    >
Comment . . . . .                               >
Multi-target change . YES                (Yes/No, Default is No)
  Target name . . . . DB2X_FILE          > (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 63. CM - Register Options panel (ADB2CRO)

3. Specify the following information in the CM - Register Options panel and issue the CONTINUE command.
 - a. Specify an owner and a name for the change. The default owner is the current SQL ID. The name of the change cannot contain an apostrophe (or single quotation mark).
 - b. Specify YES to register the change on multiple target locations. Optionally, specify a target profile.
 - c. Optionally, specify a comment for the change, whether to replace existing changes, an ignore for the change, and a mask for the change.

The change is registered as a normal change.

The Specify Register Mode panel is displayed to specify the action to be taken if pending changes are found when processing the changes from this compare job.

```

Compare ----- Generate Compare Jobs -----
Option ==>

Specify the following for DB2 Object Comparison Tool:

Worklist information:
  Worklist name . . . . . PQ76055N (also used as middle qualifier in DSNs)
Co.-----
| Compare ----- Specify Register Mode ----- 09:23
| Pending changes action . . . (Cancel, Prereq, Supersede)
|
Ch
Da
-----

```

Figure 64. Specify Register Mode (GOC5RM)

4. 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.

The change to apply the compare changes is registered.

5. After registering the change, use DB2 Administration Tool to analyze the change and then run the change to apply the changes from the comparison.

What to do next

Refer to “Managing changes to DB2 objects” in *IBM DB2 Administration Tool for z/OS User’s Guide and Reference* for information about analyzing and processing changes.

Related tasks:

“Specifying source object definitions to be compared” on page 64

The first step in defining the attributes of the DB2 objects to compare is to specify if the source object is from a DDL file, from the DB2 catalog, or from a compare version file. Use the Specify Compare Source panel to define these attributes.

“Specifying target objects to be compared” on page 88

Use the Specify Compare Target option to specify the DB2 target objects to be compared.

“Specifying the compare masks data set” on page 90

Use the Specify Compare Masks panel to specify the name of the data set that contains the translation masks. You can also use this panel to specify whether you want to edit that data set now.

“Specifying compare ignore fields” on page 95

Specify compare ignore fields when you want to ignore some fields when comparing DB2 catalog records. Use ignore fields when there are differences in source and target objects, but you don't want the compare process to change them.

Generate compare job options

Specify the criteria that you want to use to generate the work statement list or JCL for your compare job.

Use the Generate compare job panel, Figure 56 on page 106, to specify the parameters to use for the batch compare job.

Specifying worklist information

Worklist name

Specify the name of the worklist to use. This name is also used as a middle qualifier in the names of the work data sets that are created for the job.

The prefix for the work data sets is the value in the Prefix for data sets field in the Data set information section of this panel. The complete data set name is the prefix name, the worklist name, and a name that indicates the purpose of the data set. For example, for the changes file, the complete name might be NBRON.PQ76055N.CHANGES.

The worklist name is also used as a name for the DB2 Admin SQL or DDL executor, which has a checkpoint facility. The worklist name is used as a key to the checkpoint table. Use a unique name for each comparison that you run. Refer to *IBM DB2 Administration Tool for z/OS User’s Guide and Reference* for additional information about work statement lists.

Specifying compare options

Suppress DROP of objects

Specify whether the compare process will suppress dropping target objects that are in the target but that are not in the source.

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 specified for this option.

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 associated with the LOB objects is kept because Suppress DROP of objects is set to Yes, then all of the LOB objects are kept.

Suppress DROP of columns

Specify whether the compare process will suppress dropping columns that are in target tables but that are not in the source table.

Suppress adding columns

Specify whether the compare process will suppress adding columns to the target. Use this option if you have extra columns on your source system that are not on your target system.

Bypass SQL PL functions

Specify whether Object Comparison Tool bypasses the processing for SQL PL functions, including non-inline SQL scalar functions and SQL table functions:

Yes Object Comparison Tool bypasses the non-inline SQL scalar function or the SQL table function, and then continues processing for the other objects.

No Object Comparison Tool terminates processing when it detects the non-inline SQL scalar function or the SQL table function, either from the source or the target. This is the default value.

Note: When the YES option is selected, the non-inline SQL scalar function and the SQL table function are bypassed and not compared. Examine the generated APPLY job or work statement list to verify that the content is complete.

Run SQLID

Specify a valid SQL ID to use when creating, dropping, or altering objects. This is typically an administrative SQL ID, solely having the privileges to create objects. Entering <NONE> does not generate a SET CURRENT SQLID statement in the DDL. If the field is left 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).

Run Validate

The Run Validate option specifies whether consistency checking should be performed. Consistency checking verifies that the dependent objects for the primary objects in the source DDL exist. The dependent objects must exist in the source DDL or the target catalog. The following checking is performed:

- The table space in a CREATE TABLE statement exists
- The table in a CREATE INDEX statement exists
- The child and parent tables in referential constraints exist
- A clustering index does not exist if the index in a CREATE INDEX statement is a clustering index

Run Validate also checks that the dependent objects exist if the following are generated:

- CREATE TRIGGER
- CREATE VIEW
- CREATE MQT
- CREATE INDEX
- ADD FOREIGN KEY

In addition, the following checks are performed:

- For primary index and unique index changes: matching keys for primary keys and unique keys exist.
- For primary key and unique key changes: matching indexes for primary keys and unique keys exist.
- If the number of index partitions matches the number of table space partitions.

Note: For Native Stored Procedures, even if validation is successful, the existence of the object in the NSP body cannot be known at procedure run time (or during procedure call).

Validate

Specify V to indicate that consistency checking be performed. A Consistency Checks report (ADB2WVL) is generated. If a check fails, a message is written to the report with a return code of 8. Compare Target option: 4 (Target is from the DB2 Catalog and the objects are automatically selected based on the selected source objects).

None Specify N to indicate that no consistency checking be performed.

Changing reporting options

Change reporting options

Specify whether you want to change the options for reports. If you specify Yes, the Specify Compare Reporting Options panel is displayed in which you can specify your changes:

```

Compare ----- Generate Compare Jobs -----
Option ==>

Specify the following for DB2 Object Comparison Tool:

Worklist information:
Worklist name . . . . . : AAAAAAAA (also used as middle qualifier in DSNs)
Co.-----
Compare ----- Specify Compare Reporting Options ----- 12:18

Report options for Compare:
Only changed objects . . : Yes      (Yes/No)
Ch Ignore fields:
   User specified . . . . : Yes      (Yes/No)
Da   System generated . . . : Yes      (Yes/No)
   Translation masks . . . : Yes      (Yes/No)
Op   Summary report . . . . : Yes      (Yes/No)
   Object count report . . . : Yes      (Yes/No)
   Conversion report . . . . : YES      (Yes/No)

```

Figure 65. Specify Compare Reporting Options panel (GOC5RO)

- Only changed objects**
Specify whether the detailed report will include only objects that have changed.
- Ignore fields**
 - User specified**
Specify whether the user-specified ignore fields will be included in the report.
 - System generated**
Specify whether the system-generated ignore fields will be included in the report.
- Translation masks**
Specify whether the translation masks will be included in the report.
- Summary report**
Specify whether a summary report (one line per object) will be included.
- Object count report**
Specify whether an object count report will be included.
- Conversion report**
Report expected conversion problems for tables when Change runs.

The following example shows the corresponding string used in the PARM passed to step T03COMP PGM=GOC2CMP after specifying YES to a field shown on panel GOC5RO.

Specifying Yes on both User specified and System generated, under Ignore fields, will result in REPIGALL being used as the PARM.

```

Compare ----- Generate Compare Jobs -----
Option ==>

Specify the following for DB2 Object Comparison Tool:

Worklist information:
Worklist name . . . . . : AAAAAAA (also used as middle qualifier in DSNs)
Co-----
Compare ----- Specify Compare Reporting Options ----- 12:18
Report options for Compare:
Only changed objects . . : REPCHG
Ch Ignore fields:
User specified . . . . . : REPIGUSR
Da System generated . . . . : REPIGSYS
Translation masks . . . . : REPMASK
Summary report . . . . . : REPSUM
Op Object count report . . . : REPCOUNT
Conversion report . . . . : REPCONV
-----

```

Figure 66. Example of the Specify Compare Reporting Options panel (GOC5RO) with REPIGALL used as the PARM options.

Changing data set information

PDS for batch jobs

Specify the name of the partitioned data set in which the generated jobs will be created.

Prefix for data sets

Specify the prefix to be used when allocating data sets in the batch job (UNLOAD, DDL, and LOAD statement).

CHANGES file data set name

Specify the CHANGES file data set name.

CHANGES file member name

(Optional) Specify a member name for the CHANGES file.

Specifying job options

Single compare job

Specify whether all job steps will be executed in one job. The reason for using separate jobs is to run the source extraction on a system other than the target system.

- No** Up to four jobs are generated:
 - Extract source if DDL or DB2 catalog
 - Extract target if DDL or DB2 catalog
 - Compare source and target
 - Generate apply jobs or Register job when Change Management is enabled, if Generate apply jobs is set to Yes

Yes A single job is generated.

Member name

If you specify Yes in the single compare job field, you must specify a member name of the Single job. The default is COMPARE.

Generate online

The Generate online function specifies that the compare process is to be run online.

Yes Specifies that the compare process be performed online. The compare process runs immediately when the Enter key is pressed after the Generate compare options are specified.

If Generate apply jobs and As work statement list are also selected, the work statement list is created online, but is not run. To run the work statement list, access the DB2 Administration Tool Manage work statement list function.

No Specifies that a batch job will be generated instead of running the comparison online. The batch job can be submitted later to perform the compare process in the background. The default value is No.

Refer to “Running a work statement list” on page 136 for information about running work statement lists.

Use the TU command to define the ADBWORK template to override the name of the output data set that is created. This data set contains the messages that are normally placed in the SYSPRINT data set when the compare process is run in batch. If you do not specify Yes to the Generate templates option and define ADBWORK, the default data set name, *prefix.wsl.SYSPRINT*, is used. Refer to the TU option explanation for information about modifying and defining templates. Refer to *IBM DB2 Administration Tool for z/OS User's Guide and Reference* for additional information about templates.

Restriction: This function is not available with MultiCompare.

Generate apply jobs

The Generate apply jobs function generates jobs, generates a work statement list, or registers a change in the Change Management database on the target subsystem. The work statement list, or the change, or the apply jobs are then used to apply changes to the target object. This process is shown in Figure 1 on page 6.

Generating apply jobs is an optional step. You can submit the jobs or work statement list that it generates after the compare process. The generate apply jobs function uses the following input:

- The changes file from the compare operation

This file contains:

- The DROP, CREATE, and ALTER statements
- The UNLOAD requests
- Table space information records, which allow the generate apply jobs function to determine the size of the UNLOADs

Restriction: Do not attempt to import a CHANGES file generated by a normal comparison job into Change Management as a change. Doing so can lead to loss of data when the change is run.

- The shared variables file from the online dialog

This file contains the variables that were specified in the online dialog.

Yes Specify Yes to generate jobs to apply the changes or, when Change Management is enabled, to register the changes in the Change Management database. If you request the generation of apply jobs,

you must run the Compare dialog connected to the target DB2 system to pick up the correct libraries for use in the apply jobs. Alternatively, if the dialog is run on a different DB2 system, you must manually update the apply jobs to use the correct DB2 libraries.

When you specify Yes to Generate apply jobs, you also have the option to create a work statement list, use the utility options, select the unload method, and generate templates.

No Specifying No does not generate jobs to apply the changes.

Change

Specifying Change will cause DB2 Object Comparison Tool to generate a delta change in the CHANGES file suitable for later Import (that is, the CMDELTA parameter for GOC2CMP). No apply jobs or WSL are generated.

The Generate apply jobs function runs as an EXEC (ADBGJOB) in a TSO/ISPF batch job and uses ISPF skeletons to generate the apply jobs or work statement list.

Generate one job

Specify whether to generate jobs to apply the changes.

Restriction: The Generate one job options do not apply when As work statement list is set to Yes.

Yes Specify Yes to generate a single apply job. You can provide a member prefix, or you can accept the default, which is APPLY. If the number of steps exceed the limit of 255, more than one job is generated. For example, if the Member prefix is set to APPLY, the member names are APPLY001, APPLY002, and so on.

No When Generate one job is set to No, multiple jobs are created. The generate apply jobs function produces the following jobs when Generate one job is set to No:

T10Ummnn

UNLOAD jobs.

- Step 1 in these jobs issues a DB2 command to place the table space in read-only status.
- Step 2 deletes the data sets for SYSREC and SYSPUNCH, if they already exist.
- Step 3 unloads the given table. Further steps create DB2 LOAD utility control statements for the unloaded data.

The generated UNLOAD jobs can be run in parallel. The space parameter for the SYSREC data set in step 3 is derived from the RUNSTATS statistics in the catalog and from the high-used-rba value of the data set from the table space. If the target version files are not created from a DB2 catalog you must inspect and possibly correct, the space parameter because no space data is available and default sizes are used.

Note: Version files that are created outside the scope of the DB2 catalog, like DDL, do *not* represent the same information that is found in the DB2 catalog.

T20DROP

DROP job. This job drops all the objects that need to be dropped because they either were not in the target or they need to be re-created. This job contains DD statements that refer to all unload data sets to ensure that all UNLOAD jobs have run before the objects are dropped.

T30CREAT

CREATE job. This job creates the objects and their authorizations.

T40STOP

STOP job. This job is used to STOP page sets.

T50ALTER

ALTER job. Typically, this job is used to establish referential integrity constraints.

T60START/T61START

START jobs. These jobs are used to START page sets. The T61START job pertains to clone objects.

T70Rnnnn

RELOAD jobs.

T80Rnnnn

REORG jobs.

T81REBLD

REBUILD job. This job is used to rebuild indexes.

T85REFR

REFRESH job. This job is used to refresh tables.

T89POSTI

Table GRANTs and Reload Accelerator jobs

T90RB

REBIND jobs.

Utilities

After the LOAD jobs have run, optional jobs are created to run CHECK (T71CHECK), COPY (T73IMC), and RUNSTATS (T87RUNST). You can run these jobs in parallel.

(Per) Process

When the Generate one job option is set to Process, one job is created per process. For example, all UNLOAD jobs are merged into one job for each process:

UNLOADs: T10U0001

DROPs: T20DROP

CREATEs: T30CREAT

STO Cmds: T40STOP

ALTERs: T50ALTER

STA Cmds: T60START, T61START

RELOADs: T70R0001

REFRESH: T85REFR

Table GRANTS: T89POSTI

Utilities:

- CHECK: T71CHECK
- IMAGECOPY: T73IMC
- REORG: T80R0001
- REBUILD: T81B0001
- RUNSTATS: T87RUNST
- RELOAD Accelerator: T89POSTI
- REBIND: T90RB

However, if the number of steps in an UNLOAD, RELOAD, or REORG job (T10U0001, T70R0001, or T80RO001) exceed a maximum of 255, a second job corresponding to each process (T10U0002, T70R0002, or T80R0002) is generated accordingly.

When you specify Process, the Member prefix option does not apply.

Member prefix

If you specify Yes to Generate one job, provide a Member prefix or accept the default which is APPLY. This option does not apply when No or the Process option is selected.

As work statement list

Specify whether to append or replace the apply changes to a work statement list.

Yes When you specify Yes and the work statement list that you specified already exists, the Specify Work Statement List Data Set panel, shown in the following figure, is displayed:

```

Compare ----- Generate Compare Jobs -----
Option ==>

Specify the following for DB2 Object Comparison Tool:

Worklist information:
Worklist name . . . . . : PQ76055N (also used as middle qualifier in DSNs)
Co.-----
| DB2 Admin----- Specify Work Statement List Data Set ----- 12:18
| Existing Worklist
|
| Work stmt list dsn . . . WLIST.WSL
Ch Work stmt list name . . . WLIST
|
Da Existing name action . . . (Append or Replace)
| Build JCL to run work stmt list . . . (Yes/No)
Op'-----

```

Figure 67. Specify Work Statement List Data Set panel (ADB2WLDA)

Enter a data set name and a work statement list name. Indicate whether you want to append or replace the work statement list if it already exists and whether you want to build a batch job to run the work statement list.

Restriction: The replace capability is not supported if you are using the MultiCompare function to compare more than one saved dialog. Work statement lists are automatically appended with MultiCompare. To replace work statement lists for more than one dialog, you must run the comparisons individually.

The following panel displays:

```

Compare ----- Generate Compare Jobs -----
Option ==>

Specify the following for DB2 Object Comparison Tool:

Worklist information:
Worklist name . . . . . : PQ76055N (also used as middle qualifier in DSNs)
Co.-----
| DB2 Admin ----- Specify Work Statement List -----
| Existing Worklist
|
| Appending to an existing WSL may generate duplicate dataset names.
Ch To avoid this, please specify a new middle qualifier.
|
Da Middle Qualifier . . . . D5787
|

```

Figure 68. Specify Work Statement List panel (ADB27WLD)

If needed, specify a middle qualifier. The new statement will either replace the existing statement or is appended to the work statement list with the name that you specified in the Worklist name field.

When you specify Yes and the work statement list that you specified does not exist, the following panel is displayed:

```

Compare ----- Generate Compare Jobs -----
Option ==>

Specify the following for DB2 Object Comparison Tool:

Worklist information:
Worklist name . . . . . : PQ76055N (also used as middle qualifier in DSNs)
Co-----
| Compare ----- Specify Work Statement List Data Set ----- 13:48
|
| Enter/verify the following:
| Work stmt list dsn . .
|
| Ch
| Build JCL to run work stmt list           (Yes/No)
|-----
Da

```

Figure 69. Specify Work Statement List Data Set panel (GOC5WL)

If you do not specify Yes for Generate apply jobs, the Specify Work Statement List Data Set panel does not display.

If you specify Yes to build a batch job to run the work statement list, the Specify Job Parameters panel, which is shown in the following figure, is displayed. If you specify No, the batch job to run the work statement list is built.

```

ADB2W1R ----- Specify Job Parameters ----- 09:08

Enter/verify the following:
Generate one job ==> NO           (Yes, No or Per Process)
Job library PDS  ==>
Member prefix   ==> RLS1        (Prefix, max 6 chars)
Jobname = member? ==>           (Yes/No)

```

Figure 70. Specify Job Parameters panel (ADB2W1R)

Requirement: When you specify job parameters while you are building a batch job to run the work statement list, you must specify a job library PDS. In addition, the member prefix length is limited to a maximum of six characters, regardless of the number of jobs.

No If you specify No, the apply jobs are generated in a separate data set, and the Specify Data Set Name for Apply Jobs panel, shown in the following figure, prompts you for that data set name.

In either case, if the data set does not exist, it is created.

```

Compare ----- Generate Compare Jobs -----
Option ==>

Specify the following for DB2 Object Comparison Tool:

Worklist information:
  Worklist name . . . . . : PQ76055N (also used as middle qualifier in DSNs)
Co | Compare ----- Specify Data Set Name for Apply Jobs ----- 12:18 |
   | Enter/verify the following: |
   | Data Set Name ==> APPLY.DEF1045 |
Ch | |
Da | ----- |

```

Figure 71. Specify Data Set Name for Apply Jobs panel (GOC5AJ)

If you request an apply job where the source or target are from DDL input, they must include all dependent objects. Otherwise, the loss of objects or authorizations will result. The warning message that is shown in the following figure is issued when this condition occurs. You have the option to continue or to end the operation.

```

Compare ----- Generate Compare Jobs -----
Option ==>

Specify the following for DB2 Object Comparison Tool:
                                                    More:  +

Worklist information:
  Worklist name . . . . . : PQ76055N (also used as middle qualifier in DSNs)
----- DB2 Object Compare Warning -----
You have asked to generate apply jobs, but the source and / or target
objects are not being extracted from the DB2 catalog. If your extracts
do not include all dependent objects and authorizations, this may
lead to the loss of these objects and/or authorizations.

Press ENTER to continue or END to stop this operation.
F1=HELP    F2=SPLIT   F3=END     F4=RETURN  F5=RFIND   F6=RCHANGE
F7=UP      F8=DOWN    F9=SWAP   F10=LEFT  F11=RIGHT

----- YSDA -----
Please read this carefully | YSDA   Serial (tape) device : No (Yes/No)

```

Figure 72. Generate Compare Jobs warning panel (GOCGMPW)

Use customized util opts

Specify whether you want to use the options that you set for the IMAGE COPY, CHECK DATA, MODIFY, REORG/REBUILD, RUNSTATS, UNLOAD, and LOAD utilities.

Yes If you specify Yes, utility jobs and work statement lists are generated based on the utility definitions that you specified in the DB2 Administration Tool panels. Refer to the UO option explanation for more information about how to specify utility options.

No If you specify No, only default utility options are used when the work statement list or apply jobs are generated.

Refer to *IBM DB2 Administration Tool for z/OS User's Guide and Reference* for additional information about running utilities.

Unload method

Use this field to specify the method you want to use to unload data.

Unload

Specify this option to use the DB2 Unload Utility. If Parallel unload and High Performance Unload are not valid options for the current unload, the DB2 Unload Utility is automatically used and is not displayed as an option.

Parallel unload

Specify this option to use the DB2 parallel processing feature. Parallel unload cannot be used in the following situations:

- Limit key change
- Change in number of partitions
- Use of an identity col in the partitioning key
- When the version of DB2 that you are using does not support parallel unload. Parallel unload is supported in DB2 Version 7 or higher

If DB2 Object Comparison Tool determines that the operation is not eligible for a parallel unload, it uses the DB2 unload method instead. If the operation is eligible for a parallel unload, a template is used to allow DB2 to generate names for and allocate the output unload data sets.

HPU When DB2 High Performance Unload is available, you can specify the DB2 High Performance Unload tool as the unload method.

Generate templates

Specify whether you want the compare process to generate templates that will specify data sets.

If you specify Yes, templates are generated for non-utility data sets using the definitions that you specified in the DB2 Administration Tool. Refer to the TU option explanation for information about modifying and defining templates.

If you specify No, the defaults for Prefix for data sets and Worklist name apply.

If the Take an image copy or Run REORG options are Yes, the utility templates are used. Refer to *IBM DB2 Administration Tool for z/OS User's Guide and Reference* for additional information about templates.

Stop on conversion error

Specify whether you want the compare process to stop if there is a conversion error.

Attention: If there is a conversion error, the APPLY job will not be generated. An error message will be generated:

```
Compare table source(aaaaaa.bbbbb) and target(aaaaaa.bbbbb)
Column COLNAME
Conversion not supported for Col COLNAME (TIME to INTEGER)
(D)Type changed from TIME to INTEGER
(E)This type change is not supported
Tables have identical column lists
Table aaaaaa.bbbbb will be dropped
Table will be recreated
Table data conversion jobstep will not be generated
Conversion will fail because of datatype mismatch
Run stopped because conversion(s) not supported
```

Allow rotate parts

Specify whether to generate the ROTATE PARTITION or the ALTER PARTITION statement.

- Yes** Specify Yes if your table is not partitioned or if you want to generate ROTATE PARTITION for your partitioned tables. If you specify Yes, data from the rotating partitions is unloaded before the rotate takes place. You can discard this data, or to load it back into the new partitions.
- No** If you specify No, the ALTER PARTITION statement is generated. A REORG is also generated for the affected partitions. To generate the REORG, you must specify either Mandatory or All Relevant on the Run REORG/REBUILD option. This action loads the data back into the table so that you do not have to manually perform the reload. Logical and physical partitions are preserved.

Retain GENERATED ALWAYS

Whether to retain GENERATED ALWAYS for the designated column types (ROWID or ROW CHANGE TIMESTAMP).

- Yes** Retain the GENERATED ALWAYS attribute for the specified data type.
- No** Do not retain the GENERATED ALWAYS attribute for the specified data type.

IDENTITY START value

The IDENTITY value for a table column when the table is re-created.

Original

The START value from the DB2 catalog is used.

Computed

The START value is computed based upon the identity attributes of the column.

Mask ignored fields

Whether to apply masked values to ignored fields for newly added objects if the field has been masked and ignored.

- Yes** Apply the masked value.
- No** Apply the original value from source (default).

Note: This option is not applicable to ignore files provided in the Change Management "CM Register Options" panel (ADB2CRO).

Optional jobs after Reload or Alter:

Run CHECK DATA

Specify whether to generate a CHECK DATA utility job for all table spaces that are affected by the (RE)LOAD utility jobs that are generated with DB2 Object Comparison Tool.

Take an image copy

Specify whether to generate a COPY utility job for all table spaces that are affected by the (RE)LOAD utility jobs generated or by objects that are altered with DB2 Object Comparison Tool. The following options are available:

Reload

Generates a RUNSTATS utility job for all tables affected by the RE(LOAD) utility job.

Alter Generates a RUNSTATS utility job for all altered table spaces, tables, and index objects.

Both Generates a RUNSTATS utility job for all tables affected by the RE(LOAD) utility job and all altered table spaces, tables, and index objects.

None Does not generate a RUNSTATS utility job.

Run REORG/REBUILD

Specify whether to generate REORG table space utility jobs and REBUILD index utility jobs, if needed, after applying the changes from an object comparison. The purpose of this job is to make the target system operational.

Mandatory

Generate all the REORG/REBUILD jobs that are needed to remove REORG/REBUILD pending conditions and to make the data available.

All relevant

Generate all the REORG/REBUILD jobs that are needed to fully implement the effects of the changes.

For example, changing PRIQTY is registered when the table space is altered, but the new value will not be used until the table space is reorganized.

None Do not generate REORG/REBUILD jobs. This option is invalid if you specified No to Allow rotate parts.

Run RUNSTATS

Specify whether to generate a RUNSTATS utility job for all table spaces that are affected by the (RE)LOAD utility jobs generated or by objects that are altered with DB2 Object Comparison Tool. The following options are available:

Reload

Generates a RUNSTATS utility job for all tables affected by the RE(LOAD) utility job.

Alter Generates a RUNSTATS utility job for all altered table spaces, tables, and index objects.

Both Generates a RUNSTATS utility job for all tables affected by the RE(LOAD) utility job and all altered table spaces, tables, and index objects.

None Does not generate a RUNSTATS utility job.

Run REBIND

Specify whether to generate a job to rebind the plans and packages affected by changes generated by DB2 Object Comparison Tool.

Utility options

BP - Change batch job parameters

Enter BP in the command line field to access the Batch Job Utility Parameters panel, shown in the following figure. You can change batch job parameters, such as the job card, default name, space parameters, unit, and threshold on this panel. You must specify the percentage increase in space for the converted data set for the Unload PCT option. Refer to *IBM DB2 Administration Tool for z/OS User's Guide and Reference* for additional information about the Batch Job Utility Parameters panel.


```

DB2 Admin ----- DSN8 Batch Job Utility Parameters ----- 11:02
Command ==>

Generate Job Card . ==>      (Yes/No)                DB2 System: DSN8
Job cards:                  DB2 SQL ID: JS4678S
====> //JD4678SD JOB ,'DB2 UTILITY',
====> //          REGION=8M,NOTIFY=USER1,
====> //          MSGCLASS=X,
====>
====>
Generate Job CLASS ==> YES  (Yes/No)    JOB CLASS . . . . . ==>

JOBPARM:
====>
====>
====>
====>
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)
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      ==> SYSDA
Space unit . . . ==> TRK (BLK, TRK, CYL or 4096-32760)
Max Primary . . . ==> 65535 (In above units or 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 73. Batch Job Utility Parameters panel (ADB2UPA)

TU - Specify TEMPLATE usage

Enter TU in the command line field to access the Specify TEMPLATE UTILITY Usage panel from the DB2 Administration Tool:

```

DB2 Admin ----- DSN8 Specify OC TEMPLATE Usage ----- 10:12
Command ==>

Line commands:
T - Toggle Use On/Off  C - Clear data  ? - Choose Template for the Keyword
E - Edit Template
Template type          ==> OC          (UTIL, ALT, MIG, RES, OC)
Generate template statements ==> NO    (Yes/No)
Sel Keyword           Use Template Comment
-----
More: +

GOCALTR
GOCCREA
GOCDROP
GOCRBND
GOCIFFN
GOCSHVR
GOCCHNG

```

Figure 74. Specify TEMPLATE UTILITY Usage panel (ADB25TU)

Use this panel to modify the templates for DB2 Object Comparison Tool data sets. The default work data sets and descriptions are shown in the following table:

Table 8. Work data set descriptions

| Template keyword | Default data set | Description |
|------------------|-----------------------------------|-----------------------------|
| GOCALTR | <i>prefix.worklist.DDL.ALTER</i> | Primarily ALTER statements |
| GOCCREA | <i>prefix.worklist.DDL.CREATE</i> | Primarily CREATE statements |
| GOCDROP | <i>prefix.worklist.DDL.DROP</i> | Primarily DROP statements |
| GOCRBND | <i>prefix.worklist.CMD.REBIND</i> | REBIND control statements |
| GOCIFFN | <i>prefix.worklist.IFF</i> | Internal version file |
| GOCSHVR | <i>prefix.worklist.SHRVARS</i> | ISPF variables |
| GOCCHNG | <i>prefix.worklist.CHANGES</i> | Changes from compare |

The valid variables that you can specify are:

- The following functional variables:

&GOCPRE

Prefix for data sets. User provided on the Generate Compare Jobs panel Figure 56 on page 106.

&GOCWLN

Worklist name. User provided on the Generate Compare Jobs panel Figure 56 on page 106.

- The date and time variables that are supported for normal DB2 utility template processing and &USERID.

Refer to *IBM DB2 Administration Tool for z/OS User's Guide and Reference* for additional information about the Specify TEMPLATE UTILITY Usage panel and additional information about defining templates.

UO - Change utility options

Enter UO in the command line field to access the Utility Options panel:

```

DB2 Admin ----- DB2 ADMIN Utility Options ----- 10:29
Option ==>

Review / Change Options For Utility

C - Image Copy
KD - Check Data
M - Modify
O - Reorg
R - Runstats
U - Unload
L - Load

```

Figure 75. Utility Options panel (ADB2UOPS)

Select the DB2 utility to change. The DB2 Administration Tool utility options panel for your selection is displayed. Enter the options for this utility and press Enter to save the panel.

Requirement: Some utility options are not available for utility jobs that are built by DB2 Object Comparison Tool functions. You must set the Use utility options on the Generate Compare Jobs panel (Figure 56 on page 106) to Yes for DB2 Object Comparison Tool to generate utilities using these options. The options that you select are retained and used for any subsequent jobs where Use utility options is set to Yes. Refer to *IBM DB2 Administration Tool for z/OS User's Guide and Reference* for additional information about running utilities.

Related concepts:

“DB2 Object Comparison Tool components and processes” on page 5
This section provides short descriptions of DB2 Object Comparison Tool components and processes.

Related tasks:

“Running a work statement list” on page 136
When you generate a work statement list by specifying options on the Generate compare jobs panel, the statements are placed in the specified work statement library. You must then access the DB2 Administration Tool Manage work statement list function to run the work statement list.

“Generating a compare batch job” on page 105
Select the Generate compare job option to specify the parameters for generating the batch compare job.

Related reference:

Chapter 10, “Batch compare report format,” on page 171
The batch compare program produces a report that contains data based on the change reporting options selected on the Generate Compare Jobs panel.

Implicit LOB and XML table support

The DB2 Admin Tool 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)
```

For clones, this default is used:
DSN(&US..&SSID..&DB..&SN..CLONE.&UQ)

Enable authorization switching

Enabling *authorization switching* will generate DDL that is used by the authorization switching feature.

Authorization switching enables you to execute DDL and DCL under the authority of another user.

Generating a compare batch job for a multi-target import

Select the Generate compare job option to specify the parameters for generating the batch compare job. You can register the change so that the Change Management function can later import the change to objects on multiple target environments.

About this task

Prerequisite: You must specify a compare source and a compare target before you can generate a compare job.

Use the following steps to specify the options for generating a compare batch job:

1. Select option 5 on the DB2 Object Comparison Tool Menu to display the Generate Compare Jobs panel:

```

GOC5 ----- Generate Compare Jobs -----
Command ==>

Specify the following for DB2 Object Comparison Tool:

Worklist information:
  Worklist name . . . . . D24555  (also used as middle qualifier in DSNs)

Compare options:
  Suppress DROP of objects . No      (Yes/No)
  Suppress DROP of columns . No      (Yes/No)
  Suppress adding columns . . No      (Yes/No)
  Run SQLID . . . . .                (Blank, an SQLID, or <NONE>)
  Run Validate. . . . . N            (Validate, None)
  Object Grantor . . . . .           (Blank or an SQLID)
  Allow implicit DROP of
    excluded objects . . . . No      (Yes/No)

Change reporting options . . Yes      (Yes/No)
Save compare results . . . . Yes     (Yes/No)

Data set information:
  PDS for batch jobs . . . . D24555.JCL
  Prefix for data sets . . . GXRON.D24555
  Changes file data set name         (leave blank for default)
  Member name . . . . . COMPARE      (if Changes file is an existing PDS)

Options:
  Generate online . . . . . No        (Yes/No)
  Single compare job . . . . No        (Yes/No)
  Member name . . . . . COMPARE       (default COMPARE)
  Generate apply jobs . . . . Yes      (Yes, No, or (Delta) Change)
  Generate one job. . . . . Yes        (Yes, No, or (Per) Process)
  Member prefix . . . . . APPLY       (default APPLY)
  As work statement list . Yes        (Yes/No to append to work stmt list)
  Use customized util opts. Yes        (Yes/No)
  Unload Method . . . . . P           (Unload, Parallel unload, HPU)
  Generate templates. . . . No         (Yes/No)
  Stop on conversion error. NO         (Yes/No)
  Use DEFER YES . . . . . YES         (Yes/No)
  Allow rotate parts . . . . YES       (Yes/No)
  Retain GENERATED ALWAYS:
    For ROWID . . . . . YES           (Yes/No)
    For ROW CHANGE TIMESTAMP. YES      (Yes/No)
  Retain sequence:
    START and RESTART values. NO       (Yes/No)
    IDENTITY START value . . . ORIGINAL (Original, Computed)
    Mask ignored fields. . . . NO      (Yes/No)

  Optional jobs after Reload or Alter:
    Run CHECK DATA . . . . Yes        (Yes/No)
    Take an image copy . . R           (after: Reload/Alter/Both/None)
    Run REORG/REBUILD . . . M         (Mandatory, All relevant, None)
    Run RUNSTATS . . . . . R          (after: Reload/Alter/Both/None)
    Run REBIND . . . . . Yes          (Yes/No)

BP - Change batch job parameters
TU - Specify TEMPLATE usage
UO - Customize utility options

```

Figure 76. Generate Compare Jobs panel (GOC5)

- Specify the values for each of the options, and specify No for the **Generate Online** option. Press Enter.

The Change Management Prompt is displayed.

- Select Change Management for this function and press Enter.

The Register Options panel is displayed as shown in the following figure.

```

ADB2CRO n ----- CM - Register Options ----- 16:25
Command ==>

Commands: CONTINUE                               DB2 System: DB2X
                                                DB2 SQL ID: GX090

Specify the following values to register a change:

Owner . . . . . GX090                > (Optional, Default is GX090)
Name . . . . . TEST2                    >
Comment . . . . .                               >
Multi-target change . YES                (Yes/No, Default is NO)
  Target name . . . . PSVTEST            > (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 77. CM - Register Options panel (ADB2CRO)

- Specify Yes in the **Multi-target change** field and issue the Continue command. The Associate Targets panel is displayed as shown in the following figure.

```

DB2 Admin ----- CM - Associate Targets ----- Row 1 to 1 of 1
Command ==>                                         Scroll ==> CSR

Details for multi-target change: PEDRO.TEST1       DB2 System: DSNA
                                                DB2 SQL ID: PEDRO

Commands: NEXT

  Target
Sel Name  DB2 Location  Change  Change  Status
  *       *            *      name   *
----->----->----->----->----->----->----->
  PSVTEST DBAD                               NEW
***** END OF DB2 DATA *****

```

Figure 78. CM - Associate Targets panel (ADBPCMT)

- Issue the NEXT command to build and show the compare job.

Specifying masks for a multi-target import

When you import changes from a compare job to objects on multiple target environments, you can specify masks for the compare job on the following panels:

GOC3 Specify masking on this panel if your compare source and target object use different naming conventions.

ADB2CRO

Specify masking on this panel if your multi-target change to uses different naming conventions than your compare target.

ADBPCMTU

Specify masking on this panel if your multiple target systems use different naming conventions than your multi-target change.

The masking you specify on one panel will not override the masking you specify on another panel. All instances of specified masking are applied.

Updating an associated target

You can manage targets and update any associated targets details.

About this task

To update a target:

Procedure

1. Specify 5, Generate compare job on the Object Comparison Tool Menu.
2. Specify any value on the Generate Compare Jobs panel and press Enter.
3. Specify Yes on the Change Management Prompt panel to indicate that you wish to use Change Management.
4. Specify CONTINUE on the Register Options panel. If the Multi-target change field is set to YES, panel ADBPCMT is displayed. If NO is specified, the compare job is built immediately.
5. Specify the U line command next to the target you wish to update. The Update Associated Target panel is displayed, as shown in the following figure:

```
ADBPCMTU n ----- CM - Update Associated Target ----- 10:05
Command ==>

Press Enter to confirm changes.

Name . . . . . : PSVTEST
DB2 location . . . . . : DBAD
Target Defaults:
Mask owner . . . . . : USER02
Mask name . . . . . : MASKUSR2 >
Target Overrides:
Change owner . . . . . PSV01 >
Change name. . . . . TEST2 >
Mask owner . . . . . USER02 >
Mask name . . . . . MASKUSR2 >
```

Figure 79. CM - Update Associated Target panel (ADBPCMTU)

6. Specify the override values for associated target change owner, associated target change name, target mask owner & target mask name.

The Name, DB2 location, Mask owner, and Mask name are shown for reference but cannot be updated.

Name The name of the target profile.

DB2 location

The location of the DB2 system.

Mask owner

The owner of the default mask.

Mask name

The name of the default mask.

Change owner

The Owner of the change on the target system. If left blank, the field defaults to the DB2 authorization ID when the change is registered on the target system.

Change name

The name of the change on the target system. If left blank, the field defaults to `AUTO:timestamp'` when the change is registered on the target system.

Mask owner

The owner of the mask associated with the change.

Mask name

The name of the mask.

7. Press Enter to process the update. When the Enter key is pressed, the changes are saved for use in the next step and the panel is dismissed, returning to panel ADBPCMT. The Change Owner and Change Name will have the values that were entered in panel ADBPCMTU. If PF3 is pressed, the values are not saved.

Running a work statement list

When you generate a work statement list by specifying options on the Generate compare jobs panel, the statements are placed in the specified work statement list library. You must then access the DB2 Administration Tool Manage work statement list function to run the work statement list.

About this task

Refer to *IBM DB2 Administration Tool for z/OS User's Guide and Reference* for additional information about the DB2 Administration Tool Manage work statement list function.

When no UNLOAD, RELOAD, or REORG operations are required, or when a single UNLOAD, a single RELOAD, or a single REORG operation is required, only a single job is generated for the work statement list.

When multiple UNLOAD, RELOAD, or REORG operations are required, and you want to generate a single apply job for all operations rather than a separate job for each operation, you must specify that a single apply job be generated in the Manage work statement lists function in the DB2 Administration Tool.

Note: The SYSREC TEMPLATE statement in the LOAD step is used when you select HPU for the unload process. Because it is not known if HPU will be used as the unload method before the work statement list is run in the DB2 Administration Tool, the template might seem to be unneeded when the work statement list is generated. However, the SYSREC template statement is used if you select HPU as the unload process when you build the job on the DB2 Administration Tool.

To generate single or multiple apply jobs for all operations:

1. Access the DB2 Administration Tool Manage work statement lists function.
2. Specify the work statement list DSN and the work statement list name.
The Work Statement List Library panel (ADB2W1) is displayed.
3. Specify the option to use to run the work statement list. Issue the command Options. In the Work statement list options pane, in the **Run execution mode** field, enter `online` or `batch`.
You are returned to the previous panel. You can then issue the Run command.
The HPU Prompt panel (ADB2WHPU) is displayed.
4. Specify if you want to use High Performance Unload as the unload method and press Enter.

The Specify Job Parameters panel is displayed.

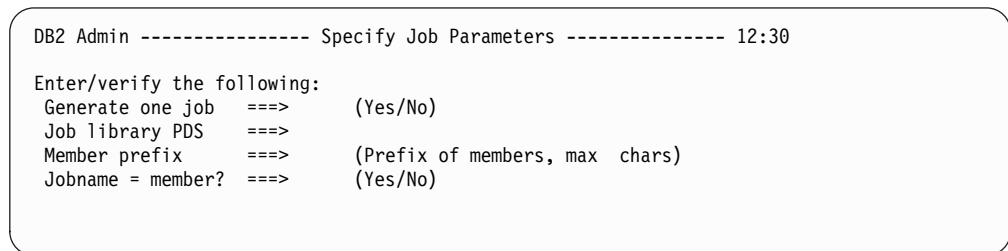


Figure 80. Specify Job Parameters (ADB2W1R)

5. Specify whether you want to generate one job by entering Yes or No in the Generate one job field.
 - a. If you specify Yes, a single batch job is generated from the work statement list that combines all the processes. The remaining fields on the panel are not used.

Table GRANT statements are generated by Object Compare after tables, related indexes, and foreign keys are created, and after all rows are reloaded.
 - b. If you specify No, enter the library name in the Job library PDS field, the member prefix name in the Member prefix field, and specify whether the jobname is the same as the member name in the Jobname = member? field.

Restriction: The maximum allowable prefix length is seven characters. When naming the member prefix, the number of characters that is required by the system to represent multiple jobs determines how many characters you can use to name the prefix. The max chars field is populated dynamically with the number of characters that are available after the number of jobs is established. For example, running 57 jobs uses two of the seven allowable characters (01, 02...57), which leaves five characters for you to use for the prefix name. Therefore, the number five is populated in the max chars field.

When you specify job parameters while you are building a batch job to run the work statement list, the member prefix length is limited to a maximum of four characters. For more information about building a batch job to run the work statement list, refer to the description of the As work statement list field.

The process generates multiple jobs and saves them in the Job library PDS named in the Specify Job Parameters panel, Figure 80.

You must submit these jobs in the sequence that they are presented in the following table. You provide the member prefix on the Specify Job Parameters panel. The remaining job name is generated based on the number of jobs being run. The examples used in the table reflect a batch that exceeds 100 jobs.

Table 9. Work statement list submission sequence for multiple operations. The following table shows examples of different jobs and what action is taken.

| Job details | Job action |
|---|--|
| <pre>prefixU001 prefixU002 prefixUnnn</pre> | One job is created for each UNLOAD operation. |
| <pre>prefix2</pre> | One job is created for all DROP, CREATE, and ALTER operations. |
| <pre>prefixL001 prefixL002 prefixLnnn</pre> | One job is created for each RELOAD operation. |
| <pre>prefix3</pre> | One job is created for all utilities except REORG. |
| <pre>prefixR001 prefixR002 prefixRnnn</pre> | One job is created for each REORG operation. |

Related reference:

“Generate compare job options” on page 113

Specify the criteria that you want to use to generate the work statement list or JCL for your compare job.

Saving dialogs

Use the Save dialog option to store the current batch job selections, called a *dialog*, for later retrieval.

About this task

Attention: The Save dialog function does not save the options that you specified in the Generate compare jobs panel, Figure 56 on page 106. You must respecify the options on that panel when you create subsequent jobs.

To save a dialog:

1. Specify option S on the Object Comparison Tool menu.
The Save dialog panel is displayed:

```
Compare ----- Save Dialog ----- 10:35
Enter/verify the following:
Prefix    ==> NBRON.ADMIN.SAVEDLGS.APAR.B37
Name      ==>
Description ==>
```

Figure 81. Save Dialog panel (ADB2SDS)

2. Specify a unique qualified name in the Prefix field for a collection of saved dialogs. This name is used as a prefix for one or more data sets in which saved dialogs are stored.
3. Specify a name in the Name field that identifies the dialog within the collection of dialogs that is identified by the prefix. If you use a duplicate dialog name within the prefix, the existing dialog is replaced. Otherwise, a new member is created.
4. Enter a description of the dialog.
5. Press Enter to save the dialog.

Managing and restoring dialogs

Use the Manage/Restore Dialogs panel to restore, rename, and delete previously saved dialogs.

About this task

To access a saved dialog:

1. Specify M to display the Saved Dialogs panel:

```
Compare ----- Saved Dialogs ----- 14:05
Option ==>

Enter the prefix for saved dialog data sets:
Prefix ==> NBRON.ADMIN.SAVEDLGS.APAR.B37
```

Figure 82. Saved Dialogs panel (GOCMC1)

2. Enter the prefix for the saved dialog data set and press Enter.
The Manage/Restore Dialogs panel is displayed, which lists all the saved dialogs in the selected prefix:

```

DB2 Admin ----- Manage/Restore Dialogs ----- Row 1 of 2
Command ==>>>                                     Scroll ==>> PAGE

Line commands: S - Select D - Delete R - Rename

S Name      Description                               Created      Id
*           *                                       *           *
-----
TEST01     SOURCE IS DDL W/ADD.COLUMN          2003/07/09  NBRON
PREVTEST   DDL&DB2 W/STOGRROUP                2003/09/09  NBRON

```

Figure 83. Manage/Restore Dialogs panel (ADB2SDM)

3. Enter the line command that corresponds to the function you want to perform. The following line commands are available:
 - S** Select a dialog to restore. Enter S in the line command field to restore the selections for the corresponding dialog. You can immediately run the batch job.
 - D** Delete a dialog. Enter D in the line command field to delete the corresponding dialog.
 - R** Rename a dialog. Enter the new dialog name in the Member Rename dialog box that is displayed when you issue the R command. Press Enter to confirm the rename.
4. Press Enter to initiate the command.

Comparing multiple sources and targets

Use the MultiCompare option to compare one or more saved dialogs.

About this task

The following exceptions apply when processing multiple compare jobs:

- If the Single compare job option is Yes and the number of steps in the job exceeds 255, the member name is truncated if it exceeds six characters to allow the addition of a numeric suffix.
- If the Single Compare job option is No, the member name is truncated if it exceeds five characters to allow the addition of a numeric suffix to indicate the compare job for each target.
- The As work statement list option must be set to Yes for MultiCompare.
- The option to replace work statement lists is not supported when generating apply jobs. Work statement lists are automatically appended. To replace work statement lists for more than one dialog, you must run the jobs individually.
- The work statement list name is derived as *Work List Namennn*. The *nnn* suffix uniquely identifies the work statement list for each target.
- Generating a work statement list online is not supported with MultiCompare. When the Generate Compare Jobs panel is displayed when specifying options during the MultiCompare process, the generate online option field value is automatically set to No.

To compare one or more saved dialogs:

1. Specify option MC on the Object Comparison Tool menu panel.
The Saved Dialogs panel is displayed:

```

Compare ----- Saved Dialogs ----- 14:05
Option ==>

Enter the prefix for saved dialog data sets:
Prefix ==> RAJESHR.ADMIN.SAVEDLGS

```

Figure 84. Saved Dialogs panel (GOCMC1)

2. Enter the prefix of the data set where the dialogs are saved and press Enter. The Manage Dialogs MultiCompare panel is displayed, which contains a list of saved dialogs:

```

Compare ----- Manage Dialogs MultiCompare -----
Command ==>                               Scroll ==> PAGE

Commands:  RUN  RUNALL

Line commands: S - Select    D - Delete    I - Interpret
               RL - Repeat/change location  Rnn - Repeat 'nn' times
               MS - Modify Source           MT - Modify Target
               IC - Include in Compare      XC - eXclude from Compare

S   Name      Location      Description Created   Id      I Status
----->-----
TARGET1 STPLEX4A_DSN8  AQ238S1 ON  2004/08/13 RAJESHR  Y Incomplete

```

Figure 85. Manage Dialogs MultiCompare (GOCMC)

The following fields are available:

S Issue a line command in the S field that corresponds to the dialog.

Name The name of the dialog.

Location

The location where the target points if DB2.

Description

The description of the saved dialog.

Created

The date the dialog was created.

Id

The user ID of the person who created the dialog.

I (Indicator)

Indicates (Y or N) if the dialog is to be included in the compare when the RUN or RUNALL commands are issued. This field is set by using the IC and XC commands.

Status

Indicates the status of the dialog. An Incomplete status means that not all source and target definitions are provided for that dialog and the dialog will not be included in the compare even if the indicator is set to Y.

3. Select the dialogs that you want to compare and issue one of the following commands:

RUN Issue the RUN command to run the compare process for all dialogs selected using the IC line command.

RUNALL

Issue the RUNALL command to run the compare process for all the dialogs except those excluded using the XC line command.

Attention: RUN and RUNALL will fail if any of the dialogs have an incomplete status.
The MultiCompare Select panel is displayed when the RUN or RUNALL command is issued:

```
Compare -----MultiCompare/Select Dialog ----- 14:18
Selecting this option enables the compare process to use the source of the
following dialog to be compared against all the targets selected.

Compare one source to multiple targets? ==> (Yes/No)
Dialog Name                               ==> <Dialog Name>
```

Figure 86. MultiCompare/Select (GOC2MCMC)

- Specify whether you want to compare one source to multiple targets.
To compare one source to multiple targets, type Y and the dialog name that contains the source to be used, and press Enter. The source details of the selected dialogs are used in each compare process instead of the source in the saved dialog.
To compare using the source and target that are defined in each saved dialog, enter N. The current source details that are specified for each of the selected dialogs are used. The Generate Compare Jobs panel is displayed:

```

GOC5 ----- Generate Compare Jobs -----
Command ==>

Specify the following for DB2 Object Comparison Tool:

Worklist information:
  Worklist name . . . . . PQ76055N (also used as middle qualifier in DSNs)

Compare options:
  Suppress DROP of objects . No          (Yes/No)
  Suppress DROP of columns . No          (Yes/No)
  Suppress adding columns . No          (Yes/No)
  Run SQLID . . . . .                   (Blank, an SQLID, or <NONE>)
  Run Validate. . . . . V                (Validate, None)
  Object Grantor . . . . .              (Blank or an SQLID)
  Allow implicit DROP of
    excluded objects . . . . N          (Yes/No)
  Enable auth-switching . . . YES       (Yes/No)

Change reporting options . . Yes        (Yes/No)
Save compare results . . . . Yes       (Yes/No)

Data set information:
  PDS for batch jobs . . . . CMP.PQ76055N
  Prefix for data sets . . . NBRON
  Changes file data set name.
    Member name . . . . .                (if Changes file is an existing PDS)

Options:
  Generate online . . . . . Yes          (Yes/No)
  Single compare job . . . . No          (Yes/No)
    Member name . . . . . COMPARE       (default COMPARE)
  Generate apply jobs . . . . Yes        (Yes, No, or (Delta) Change)
    Generate one job. . . . . Yes       (Yes, No, or (Per) Process)
    Member prefix . . . . . APPLY       (default APPLY)
  As work statement list . Yes           (Yes/No to append to work stmt list)
  Use customized util opts. Yes          (Yes/No)
  Unload Method . . . . . : P           (Unload, Parallel unload, HPU)
  Generate templates. . . . No          (Yes/No)
  Stop on conversion error. NO           (Yes/No)
  Use DEFER YES . . . . . YES          (Yes/No)
  Allow rotate parts . . . . YES         (Yes/No)
  Retain GENERATED ALWAYS:
    For ROWID . . . . . YES             (Yes/No)
    For ROW CHANGE TIMESTAMP. YES       (Yes/No)
  Retain sequence:
    START and RESTART values. NO        (Yes/No)
  IDENTITY START value . . . ORIGINAL   (Original, Computed)
  Mask ignored fields . . . . NO        (Yes/No)

Optional jobs after Reload or Alter:
  Run CHECK DATA . . . . Yes           (Yes/No)
  Take an image copy . . R              (after: Reload/Alter/Both/None)
  Run REORG/REBUILD . . . M            (Mandatory, All relevant, None)
  Run RUNSTATS . . . . . R             (after: Reload/Alter/Both/None)
  Run REBIND . . . . . Yes             (Yes/No)

BP - Change batch job parameters
TU - Specify TEMPLATE usage
UO - Customize utility options
CO - Change options common to change functions

```

Figure 87. Generate Compare Jobs panel (GOC5)

Set the values to use for running this compare job. Refer to “Generate compare job options” on page 113 for definitions of the options available on the Generate Compare Jobs panel.

5. Enter the line command that corresponds to the function that you want to perform.

In addition to Select and Delete, the following line commands are available:

I - Interpret

Displays the Interpret Dialog panel which contains the source and target details for the corresponding dialog:

```

Compare ----- Interpret Dialog ----- Row 1 to 11 of 11
                                           Scroll ==> PAGE

-----
Type          Name
-----
Mask data set  None specified
Ignore data set Using defaults
Source Version data set D3410.VER.DSN
Location      STPLEX4A_DSN7
Table space   VNDRJPDB.VNDRJPTS
Table space   AHXFLWDB.AHX3UJWU
Table space   ADBD4BAS.ADBS4BAS
Table space   ADBDCH3.ADBSCH3
Target Version data set D3410.TGT.VER
Location      STPLEX4A_DSN7
Table space   VNDRJPDB.CQ289TS
  
```

Figure 88. Interpret Dialog (GOCMI) panel

This panel indicates whether the source or target is a DDL or DB2 catalog. If the source or target is a DB2 catalog, this command lists all objects in the catalog and their type.

RL - Repeat/change location

Use the RL command to replicate a dialog and change the location of the target. Issue the RL in the Select field that corresponds to the dialog to copy and press Enter. The Distributed DB2 Systems panel is displayed:

```

DB2 Admin ----- Distributed DB2 Systems -----
Command ==>                                           Scroll ==> PAGE

Select by typing '+'
Select the location you wish to use:                   DB2 System: DSN8
                                                       DB2 SQL ID: RAJESHR

Line commands:
  S - Use DDF to access remote catalog  CO - Connect to remote subsystem
  DIS - Display threads for remote system

Select Location
-----
  STPLEX4A_DSN7
  SQLVM6
  STLEC1
  
```

Figure 89. Distributed DB2 Systems panel (ADB2DDF)

The Distributed DB2 Systems panel displays the remote DB2 subsystems that are available from the DB2 subsystem that you are currently on. Select the new location or locations for the dialog by entering a plus sign (+) in the Select field. You can select multiple locations at one time. When you press End, the Repeat Dialog/Change location panel is displayed, which contains the new dialog with the new location:


```

Compare----- Repeat Dialog/Change Location --- Row 1 to 1 of 1
Command ==>                                     Scroll ==> PAGE

Commands: CONTINUE

Specify output compare version file:
Data set prefix: J148286.OC                (Prefix of target version files)
Data set suffix: TGTVF                    (Optional suffix)

S Target Location Dialog Dialog Description
* * *
-----
STPLEX4A_DSN8 TEST1

```

Figure 90. Repeat Dialog/Change Location (GOCMCRL)

You can select and edit the dialog name and description. When you are finished with the edits, issue the Continue command to return to the Manage Dialogs/MultiCompare panel.

Rnn - Repeat nn times

Use the Rnn command to replicate a dialog multiple times. The Repeat Dialog/Change location panel is displayed which contains the replicated dialogs when this command is invoked. You can edit the dialog target location, name, and description. When you are finished with your edits, issue the Continue command to return to the Manage Dialogs/MultiCompare panel.

MS - Modify Source

Use this command to access the Specify Compare Source panel, Figure 12 on page 65, to modify the source details.

MT - Modify Target

Use this command to access the Specify Compare Target panel, Figure 37 on page 88, to modify the target details.

IC - Include in Compare

Selects a dialog to include in the compare process. Upon selection, the Indicator field is set to Y. Dialog status must be *Complete* for it to be included in the compare process or an error will occur.

XC - Exclude from Compare

Selects a dialog to exclude from the compare process when the RUNALL command is issued. You can also use this command to reverse the IC command and change the indicator field from Y to N.

Related tasks:

- “Specifying source object definitions to be compared” on page 64
- The first step in defining the attributes of the DB2 objects to compare is to specify if the source object is from a DDL file, from the DB2 catalog, or from a compare version file. Use the Specify Compare Source panel to define these attributes.
- “Specifying target objects to be compared” on page 88
- Use the Specify Compare Target option to specify the DB2 target objects to be compared.

Chapter 5. Batch DDL file extraction program

The DDL file extraction program interprets a source file of DDL statements that define DB2 objects. The program generates an output file, called a *version file*, that contains records that are similar in format to those in the DB2 catalog that defines the same objects.

To effectively compare the input DDL objects to different versions of the same objects, you can use the version file as input to the batch Compare program.

Restriction: Version files are compressed internally and should not be created with DFSMS compression because GEN and the DDL reader opens them for update, which is not allowed for DFSMS compressed data sets. DB2 Administration Tool or DB2 Object Comparison Tool jobs will receive S213-C8 abends if the version file data sets are defined with DFSMS compression.

The batch DDL file extraction program is run and a report is produced when you set source DDL file definitions on the Specify Source DDL panel (Figure 13 on page 66) and run a compare job.

The source of the DDL statements can be:

- A sequential file that contains SQL statements
- An extract from a DB2 catalog of some set of DB2 objects and dependencies

The following topics provide additional information:

- “Supported SQL statements”
- “Batch DDL file extraction program report format” on page 150

Related concepts:

“DB2 Object Comparison Tool components and processes” on page 5

This section provides short descriptions of DB2 Object Comparison Tool components and processes.

Chapter 7, “Batch compare program,” on page 157

The batch compare program is run when you specify options on the Generate Compare Jobs panel and generate a compare batch job. This program compares two sets of DB2 objects, reports all differences, and writes all changes to a file. This file is used to generate updates to upgrade target objects to the level of source objects.

Related tasks:

“Specifying that the source is a DDL file” on page 65

Select the Specifying source DDL file definitions option to specify the data set name of the DDL file that contains the objects to compare, the data set name of the version file in which you want to put the definition of these objects, and an optional description of the objects that you are comparing.

Supported SQL statements

The DDL file extraction program supports a subset of the SQL statements that are supported by DB2 for z/OS.

DDL statements that are submitted for processing by the DDL file extraction program must be in the format that is supported by SPUFI or DSNTEP2:

- Input must be in columns 1-72.
- Phrases can span records. For example, column 1 of an input record immediately follows column 72 of the previous record.
- Comments can be included and are indicated by two consecutive dashes (--).
- 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.

Restriction: The DDL reader does not communicate with DB2. Therefore, the DDL reader is unable to acquire defaults that are established by the customer for table space buffer pool, compression and index buffer pool, and pad index. The defaults that are used are those used before DB2 Version 9.

The following SQL statements are supported:

- ALTER DATABASE
- ALTER FUNCTION
- ALTER INDEX
- ALTER PROCEDURE
- ALTER SEQUENCE
- xALTER STOGROUP
- ALTER TABLE

Restriction: ALTER TABLE ROTATE PARTITION restrictions are:

- The maximum number of ALTER TABLE statements that can be processed to rotate partitions is $n-1$, where n is the number of partitions.
- If a rotate has taken place and if new partitions have also been added, the rotate will not be detected.
- If a rotate has taken place and alter of limitkeys has also been done, the rotate might not be detected.

Restriction: Constraint names are not compared (and differences not reported) because constraint names can be explicitly specified or, if they are not explicitly specified, be generated by DB2. If the constraint names are generated by DB2, the constraint names could be different between source and target, even if the DDL for the object might be the same for source and target.

Restriction: The ALTER statement is not supported for auxiliary tables.

- ALTER TABLESPACE
- COMMENT ON

Restriction: The COMMENT ON statement is not supported for auxiliary tables.

- COMMIT
- CREATE ALIAS
- CREATE AUX TABLE
- CREATE DATABASE
- CREATE DISTINCT TYPE
- CREATE FUNCTION
- CREATE INDEX
- CREATE PROCEDURE

- CREATE SEQUENCE
- CREATE STOGROUP
- CREATE SYNONYM
- CREATE TABLE

Restriction: The LIKE form of CREATE TABLE is not supported.

Restriction: Constraint names are not compared (and differences not reported) because constraint names can be explicitly specified or, if they are not explicitly specified, be generated by DB2. If the constraint names are generated by DB2, the constraint names could be different between source and target, even if the DDL for the object might be the same for source and target.

- CREATE TABLESPACE
- CREATE TRIGGER
- CREATE VARIABLE

If a dependent object such as a procedure (native stored procedure), PL/SQL function, Trigger, View, Column mask, or Row permission in the data set references a global variable, then the CREATE VARIABLE statement should be included in the DDL data set.

- CREATE VIEW
- DROP ALIAS
- DROP DATABASE
- DROP DISTINCT TYPE
- DROP INDEX
- DROP SEQUENCE
- DROP SPECIFIC FUNCTION
- DROP STORED PROCEDURE
- DROP SYNONYM
- DROP TABLE
- DROP TABLESPACE
- DROP TRIGGER
- DROP VARIABLE
- DROP VIEW
- GRANT collection privileges
- GRANT database privileges
- GRANT distinct type or JAR privileges

Restriction: The GRANT USAGE ON JAR statement is not supported in change management, or in the DB2 Object Comparison Tool.

- GRANT function or procedure privileges
- GRANT package privileges
- GRANT plan privileges
- GRANT schema privileges
- GRANT sequence privileges
- GRANT system privileges
- GRANT table or view privileges
- GRANT use privileges
- GRANT variables

- LABEL ON

Restriction: For objects that exist on both the source and the target, DB2 Object Comparison Tool compares and reports the authorization differences, but does not propagate the differences from the source to the target. DB2 Object Comparison Tool does not propagate the differences in order to avoid corrupting the target authorizations. During the apply job, the GRANT statements from the source are ignored and the GRANT statements from the target are read.

Restriction: The LABEL ON statement is not supported for auxiliary tables.

- RENAME INDEX

The DDL of the index must be included in the source DDL along with the RENAME INDEX statement.

Restriction: Rename of an implicit index is not supported.

- RENAME TABLE
- SET CURRENT PATH
- SET CURRENT SQLID

Batch DDL file extraction program report format

The report that the batch DDL file extraction program produces begins with a header and the IBM copyright statement. The copyright statement is followed by a line that indicates the version of DB2 startup parameters that are used when the extraction program is processing statements from the input stream.

Next, if the first statement in the input stream is not a SET CURRENT SQLID statement, the program indicates the authorization ID under which the input statements are being processed. This authorization ID serves as the *owner* of objects that are created and as the default schema name when a schema name is required but not specified. The authorization ID remains in effect until it is changed by a subsequent SET CURRENT SQLID statement.

Finally, a statistical summary of the process is produced that indicates the number of:

- DDL input records in the input stream
- Unique DDL statements within those records
- Catalog records written to an intermediate data set
- Catalog records written to the final output data set

The following figure shows sample output.

GOC2DTC - Create Version File from DDL File

2006-06-09 18:57

DB2 Object Comparison Tool
5697-L40 (C) Copyright IBM Corporation 2001, 2007.
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.

Using DB2 DECP Version 8(new function mode) startup parameters for SSID DSN8

Processing under auth_id of current task, VNDR230, until changed by SET CURRENT SQLID statement.

GOC2DTC - Create Version File from DDL File

2006-06-09 18:57

GOC2DTC - Summary

| | | |
|--|---|-----|
| Number of DDL input records | : | 369 |
| Number of DDL statements | : | 41 |
| Number of Catalog records intermediate | : | 59 |
| Number of Catalog records written | : | 59 |

GOC2DTC - Successful completion

Figure 91. CREATE VERSION report from DDL file

Chapter 6. Batch DB2 catalog extraction program

The batch DB2 catalog extraction report is produced when you set target DDL file definitions on the Specify Compare Target panel and run a compare job.

This report is generated by using the DB2 Administration Tool ADB2GEN program. Refer to *IBM DB2 Administration Tool for z/OS User's Guide and Reference* for additional information about the ADB2GEN program.

The use of ADB2GEN in the compare process is controlled by two program parameters, which are set in the JCL:

- WRTCAT (write catalog records, in other words, create a version file).

Restriction: Version files are compressed internally and should not be created with DFSMS compression because GEN and the DDL reader opens them for update, which is not allowed for DFSMS compressed data sets. DB2 Administration Tool or DB2 Object Comparison Tool jobs will receive S213-C8 abends if the version file data sets are defined with DFSMS compression.

Restriction: If LOB objects are involved, a new version file layout is created. This new version file is not compatible with old version files containing LOBs. The old version files with LOBs must be regenerated.

- NOGEN (do not create DDL for extracted objects)

You specify options and object extract requests in exactly the same manner as in ADB2GEN. However, when you extract objects for DB2 Object Comparison Tool, you generate all parameter and request input by using the ISPF panels.

The following topics provide additional information:

- “Batch DB2 catalog extraction program report”

Related tasks:

“Specifying target objects to be compared” on page 88

Use the Specify Compare Target option to specify the DB2 target objects to be compared.

Batch DB2 catalog extraction program report

The report that the batch DB2 catalog extraction program produces begins with a header and the IBM copyright statement. The copyright statement is followed by a line that indicates the version of DB2 startup parameters that are used when the extraction program is processing statements from the input stream.

The report contains four parts:

- Header and IBM copyright statement
- DB2 system ID and version, followed by a summary of the parameters
- A summary of object extract requests and related messages
- A count of the number of catalog records written

The following figure shows sample output.

ADB2GEN - Create DDL from catalog info 2006-06-09 18:57

Database 2 Administration Tool
5697-L90 (C) Copyright IBM Corporation 1998, 2006.
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.

ADB2GEN - Create DDL from catalog info 2006-06-09 18:57

Input prepared by Sqlid VNDR230 on DSN8 (DB2 version 810) for use on DB2 version 810 system
Object definitions extracted from DSN8 (DB2 version 810)

Parameters for this run :

| | | | | | |
|------------------------|-------|-----------------------|-------|--------------------------|-------|
| Create Database(s) | : No | Create Tablespace(s) | : Yes | Create Table(s) | : Yes |
| Create View(s) | : Yes | Create Index(es) | : Yes | Create Synonym(s) | : Yes |
| Create Alias(es) | : Yes | Create Label(s) | : Yes | Create Comment(s) | : Yes |
| Create Triggers | : Yes | Create Foreign key(s) | : Yes | also for refs not gen'd | : Yes |
| Create User def. Types | : No | Create Functions | : No | Create Stored Procedures | : No |
| Create Sequences | : No | | | | |

| | | | | | |
|--------------------------|-------|---|-------|-------------------------|-------|
| Copy Stogroup Grant(s) | : Yes | | | | |
| Copy Database Grant(s) | : Yes | Copy Tablespace Grant(s) | : Yes | Copy Table Grant(s) | : Yes |
| Copy View Grant(s) | : Yes | Copy authorisations on referenced schema(s) | | | : No |
| Copy U.def type Grant(s) | : No | Copy Function Grant(s) | : No | Copy Procedure Grant(s) | : No |
| Copy Sequence Grant(s) | : No | | | | |

Insert COMMIT statement after every definition
RE will generate all parameters even if they take default values

ADB2GEN - Create DDL from catalog info TABLESPACE TTT8S81D FROM CAT 2006-06-09 18:57

Generating DDL for Tablespace DSN8S81D In Database TTT8D81A

ADB2GEN - Create DDL from catalog info TABLESPACE TTT8S81D FROM CAT 2006-06-09 18:57

ADB2GEN - Summary of catalog records written

Number of catalog records written : 118

ADB2GEN - Ended normally

Figure 92. CREATE VERSION report from DB2 catalog

ADB2GEN gets the DECIMAL=COMMA/PERIOD (and other DB2 parameters) from a DSNHDECP module which ADB2GEN looks for in the STEPLIB data sets. The values that ADB2GEN finds in this module might not match what DB2 is currently using, or match the values that were used to store data in catalog rows; if the modules don't match, ADB2GEN might produce incorrect DDL.

You can determine the DSNHDECP parameters that ADB2GEN is using by referring to the DSNHDECP parameter section of the ADB2GEN output listing. An example is highlighted in the following figure.

Database 2 Administration Tool
 5697-L90 (C) Copyright IBM Corporation 1998, 2006.
 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.

Input prepared by Sqlid SINNOTT on DB8A (DB2 version 810) for use on DB2 version 810 system
 Object definitions extracted from DB8A (DB2 version 810)

DB2 DSNHDECP values for this run :

| | | | |
|---|-----------------|--|----------------|
| DB2 Version, Release and Mod Level | : 810 | Default CCSID for EBCDIC SBCS | : 00037 |
| Decimal point option | : '.' | Default CCSID for EBCDIC Mixed | : 00002 |
| Subsystem ID | : DB8A | Default CCSID for EBCDIC DBCS | : 00002 |
| Graphic for DBCS data | : No | Default CCSID for ASCII SBCS | : 00437 |
| Date format | : ISO | Default CCSID for ASCII Mixed | : 00002 |
| Time format | : ISO | Default CCSID for ASCII DBCS | : 00002 |
| Default encoding scheme | : EBCDIC | Default CCSID for UNICODE SBCS | : 00367 |
| DB2 Version 8 New Function Mode | : Yes | Default CCSID for UNICODE Mixed | : 01208 |
| | | Default CCSID for UNICODE DBCS | : 01200 |

Parameters for this run :

| | | | | | |
|------------------------|-------|-----------------------|-------|--------------------------|-------|
| Create Database(s) | : Yes | Create Tablespace(s) | : Yes | Create Table(s) | : Yes |
| Create View(s) | : Yes | Create Index(es) | : No | Create Synonym(s) | : No |
| Create Alias(es) | : No | Create Label(s) | : No | Create Comment(s) | : No |
| Create Triggers | : 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 |
| Create Sequences | : No | | | | |

| | | | | | |
|--------------------------|-------|---|-------|-------------------------|------|
| Copy Stogroup Grant(s) | : Yes | | | | |
| Copy Database Grant(s) | : Yes | Copy Tablespace Grant(s) | : Yes | Copy Table Grant(s) | : No |
| Copy View Grant(s) | : No | Copy authorisations on referenced schema(s) | | | : No |
| Copy U.def type Grant(s) | : No | Copy Function Grant(s) | : No | Copy Procedure Grant(s) | : No |
| Copy Sequence Grant(s) | : No | | | | |

Insert COMMIT statement after every definition
 RE will generate all parameters even if they take default values

Figure 93. DSNHDECP values

Chapter 7. Batch compare program

The batch compare program is run when you specify options on the Generate Compare Jobs panel and generate a compare batch job. This program compares two sets of DB2 objects, reports all differences, and writes all changes to a file. This file is used to generate updates to upgrade target objects to the level of source objects.

Refer to “Generating a compare batch job” on page 105 for more information about setting batch compare options.

The batch compare program processes two version files, one that represents the (new) source version of the objects to be compared and one that represents the (old) target version.

The batch compare program performs the following tasks:

- Applies any masks to the prefix of the source version file
- Sorts the two version files
- Compares the two version files, applying masks to all relevant names and authorization IDs before comparison and ignoring any differences that are specified in the ignore file

In addition, you can create a list of objects to be excluded from a compare process by using exclude specification. The list can be created manually or based on results from a compare results stored in a DB2 table. Refer to “Exclude objects from the compare process” on page 82 for more information about creating and using Exclude Specification.

The following topics provide additional information:

- “Compare version files” on page 158
- “Special considerations for comparing DB2 objects” on page 158
- “Changing or unloading tables with LOBs” on page 165

Refer to Chapter 10, “Batch compare report format,” on page 171 for batch compare report examples.

Related concepts:

“DB2 Object Comparison Tool components and processes” on page 5

This section provides short descriptions of DB2 Object Comparison Tool components and processes.

Chapter 5, “Batch DDL file extraction program,” on page 147

The DDL file extraction program interprets a source file of DDL statements that define DB2 objects. The program generates an output file, called a *version file*, that contains records that are similar in format to those in the DB2 catalog that defines the same objects.

Related tasks:

“Generating a compare batch job” on page 105

Select the Generate compare job option to specify the parameters for generating the batch compare job.

“Specifying that the source is a version scope” on page 80

You can compare object types by defining a version scope for objects. A version scope can contain databases, table spaces, tables, indexes, views, stored procedures,

triggers, and other objects.

Related reference:

Chapter 10, “Batch compare report format,” on page 171

The batch compare program produces a report that contains data based on the change reporting options selected on the Generate Compare Jobs panel.

Compare version files

Object Comparison Tool operates on sorted version files. The version file record prefix is the sort key. Masks are applied to the prefix of the source version file before the file is sorted. The result is that objects in the source and target version files are in the same sequence.

The following results can occur when the batch compare program attempts to match object names:

- An object was not found in the source version file.
In this case, the object is registered for deletion unless the option to keep target objects was specified through the Suppress DROP of objects field in the Generate Compare Jobs panel, Figure 56 on page 106.
- An object was not found in the target version file.
In this case, the object definition is saved to create the new object at a later stage. Masks are applied to the relevant fields before the object definition DDL is built.
- An object was found in the source and target version files.
In this case, masks are applied to the source version DB2 catalog records that describe the source object.
The objects are compared; only fields for which ignore has not been specified (explicitly or by default) are included. The possible results of the comparison are:
 - Objects are identical
 - Upgrade can be performed by altering the target object
 - Upgrade requires drop and re-create of the target objectThe differences are reported, and the actions that are required to upgrade the target version to the source version (if any) are written to the CHANGES file for use at a later stage.

Special considerations for comparing DB2 objects

You can perform most comparisons field by field, comparing the catalog records that represent the objects. However, special considerations are needed in some situations.

These situations are described in the following sections:

- “Constraint names” on page 159
- “DROP statements in the source DDL” on page 159
- “Functions” on page 159
- “Implicit and explicit objects” on page 160
- “Materialized query tables” on page 160
- “Native SQL procedures” on page 160
- “Object authorizations” on page 161
- “Online schema evolution” on page 161

- “Partitioned tables” on page 161
- “Renamed objects” on page 162
- “Special considerations for comparing DB2 objects” on page 158
- “Table columns” on page 163
- “Table drop/re-create without data conversion” on page 163
- Table 11 on page 164
- “Triggers” on page 164
- “Views” on page 165
- “ALIAS for SEQUENCE not supported” on page 165

Constraint names

Constraint names are not compared (and differences not reported) because constraint names can be explicitly specified or, if they are not explicitly specified, be generated by DB2. If the constraint names are generated by DB2, the constraint names could be different between source and target, even if the DDL for the object might be the same for source and target.

Differences in constraint name are not compared because this would cause unnecessary drop and recreate of constraints that are logically correct. Dropping and recreating constraints would put the table space in CHECK PENDING, that is, out of service. Since there are no real differences between objects, just differences in the constraint names, it might not be necessary to put the table space out of service.

DROP statements in the source DDL

All DROP statements in the source DDL are copied to the DDL that is produced during the compare process. The effect of the drop statements is the same as dropping the objects on the target before running the compare job. Data from the dropped tables is saved by generated unload utilities making it possible for you to recover data from the dropped tables manually. In addition, the corresponding RUNSTATS, IMAGECOPY, and CHECK DATA utilities are not generated even if they are requested on the Generate Compare Jobs panel, Figure 56 on page 106.

All implicitly dropped objects are found when the target catalog is available. However, if the target catalog is specified in the DDL, the DROP impact might be incorrectly reported. Data in the dropped objects that is missing from the DROP impact report is not saved by generated unload utilities. It is important that you save the dropped objects if at least one DROP statement is in the source DDL and the target catalog is unavailable. DROP statements in the target DDL are ignored. The statement sequence CREATE/DROP for the same object is invalid, the result is unpredictable.

Functions

Functions are compared based on the function signature, meaning that the function-specific name is treated as an attribute of the function, and a comparison is performed. If specific names are different, the target function definition is upgraded with the source-specific name. If you do not want the function definition upgraded, SYSROUTINES.SPECIFICNAME should be ignored.

If SQL PL functions, including non-inline SQL scalar function and SQL table function, are included in the compared objects, use the compare option **Bypass**

SQL PL functions to control how DB2 Object Comparison Tool should process the objects. When the **Bypass SQL PL functions** option is specified as NO and when the non-inline SQL scalar functions or the SQL table functions are detected, DB2 Object Comparison Tool terminates processing. Otherwise, DB2 Object Comparison Tool skips the non-inline SQL scalar functions and the SQL table functions. DB2 Object Comparison Tool then continues processing the other objects and generates the APPLY job or work statement list.

Note: Examine the APPLY job or work statement list to verify that the content is complete.

Implicit and explicit objects

DB2 Object Comparison Tool compares implicit objects from the source with implicit objects from the target and explicit objects from source with explicit objects from target. All objects from DDL source are explicit objects. If no explicit counterparts are found on the target, those objects are processed as new explicit objects which need to be added to the target. If no explicit counterparts for explicit target objects are found on the source, those objects are dropped from target.

Materialized query tables

Comparisons involving materialized query tables (MQTs) do not compare columns. Instead, only the table type is compared.

For example:

1. If the target is defined as:

```
CREATE TABLE <schema>.<mqt_name> AS (  
  SELECT * FROM SYSIBM.SYSDUMMY1 )  
DATA INITIALLY DEFERRED REFRESH DEFERRED IN <dbname>.<ts_name>;
```

2. And the source is a same-named, different columned table (it does not matter if the source had 20 more columns):

```
CREATE TABLE <schema>.<table_name> (AAAAAAD CHAR(2))  
IN <dbname>.<ts_name>;
```

3. The compare output shows:

```
SDSF OUTPUT DISPLAY XXXXXXX Jnnnnnnn DSID 110 LINE 49 COLUMNS 02- 81  
COMMAND INPUT ==>>> SCROLL ==>> CSR  
Compare table source(<schema>.<mqt_name>) and target(<schema>.<tb_name>)  
(A)Table type changed from Materialized Query Table to normal table  
Table will be altered
```

```
GOC2CMP - Ended normally
```

4. And the result is:

```
--#ADMIN PROCESS CREATE  
ALTER TABLE <schema>.<mqt_name>  
  DROP MATERIALIZED QUERY ;  
COMMIT ;
```

Native SQL procedures

DB2 Object Comparison Tool compares active and inactive versions of a native SQL procedure by comparing the options and the native SQL procedure bodies. The native SQL procedure bodies are compared the same way as the trigger bodies are compared.

Object authorizations

DB2 Object Comparison Tool handles object authorizations differently, depending on the object location:

- For objects that exist on both the source and the target, DB2 Object Comparison Tool compares and reports the authorization differences, but does not propagate the differences from the source to the target. DB2 Object Comparison Tool does not propagate the differences in order to avoid corrupting the target authorizations. During the apply job, the GRANT statements from the source are ignored and the GRANT statements from the target are read.
- For objects that exist only on the source and that are added to the target during the apply job, the source authorization is applied to the target objects.

Online schema evolution

The following DB2 Version 8 Online schema evolution functionality is not exploited by DB2 Object Comparison Tool. This means that none of the related ALTER statements are generated when applying these types of changes:

- Alter of Identity column attributes.
- Add partitioning key. This function is intended for adding partitioning information for a table in a partitioned table space if the definition of the table is incomplete.

Partitioned tables

Tables in partitioned table spaces can be dropped only by dropping the table space. If a table in a partitioned table space has changes that require the table to be dropped and re-created, the partitioned table space is dropped and re-created as well, even if the table space comparison shows no differences.

DB2 Object Comparison Tool can accept differences in the number of partitions by ignoring the field SYSTABLESPACE.PARTITIONS. In this case, no comparisons are performed at a partition level, and all partition characteristics are taken from the target.

If the table space is not part of the comparison (that is, the comparison is performed at the table level), the following conditions apply when a partitioned table needs to be dropped and re-created:

- If the target is a DDL file, the table space cannot be dropped and re-created because the table space definition is not available.
- If the target contains tables from the DB2 catalog, the table space definition from the catalog is stored in the version file. Unless otherwise indicated, the fact that a table is partitioned is derived from the stored table space definition. In any other case, the table space definition is used only for the purpose of re-creating the table space.

Pending Changes

Pending changes are included in version files created from catalog records. No pending changes can be included in version files from DDL. Pending changes are merged into the changed objects before the objects are compared. If pending changes are ignored, the source and target pending changes will not be merged into the changed objects. If the source DDL contains an ALTER with DROP PENDING CHANGES, the ALTER will be passed to the target and pending

changes in the target version file will be ignored.

Renamed objects

When comparing objects, DB2 Object Comparison Tool will take into consideration if a database, table space, table, index, or column was renamed in the source system. You can inform DB2 Object Comparison Tool when a rename has occurred in the source system by using rename specifications. Enter rename specifications the same way that you enter compare masks. For more information about entering rename specifications, refer to “Specifying the compare masks data set” on page 90. When you specify that an object or column was renamed in the source, DB2 Object Comparison Tool compares the existing object in the target with the renamed object in the source. When the target object is updated, the data in the target system is preserved. For example, you have the following source and target objects:

```
Source = CREATE TABLE USERA.T2 (COLA, COLB, COLY, COLZ)
Target = CREATE TABLE USERA.T2 (COLA, COLB, COLY, COLZ)
```

If you rename the source table T2 to T1 and COLY to COLX, the source and target objects are now different.

```
Source = CREATE TABLE USERA.T1 (COLA, COLB, COLX, COLZ)
Target = CREATE TABLE USERA.T2 (COLA, COLB, COLY, COLZ)
```

Using the following RENAME specifications (refer to “Specifying the compare masks data set” on page 90 for syntax examples and supported object types), the table is renamed during the compare process to T1 and COLY is renamed to COLX:

```
RENAMETB:USERA.T2,USERA.T1
RENAMECOL:USERA.T1.COLY,COLX
```

Note: The new table name (T1) is referenced in the RENAMECOL statement because the RENAMETB statement occurs before the RENAMECOL statement. If the RENAMECOL statement was issued first, you would reference the original table name in the RENAMECOL statement.

The following steps are generated on the target system:

- Unload the table T2 data
- Drop table T2 and create table T1
- Load the COLY data from table T2 data into COLX in table T1

Restriction 1: It is not always possible for DB2 Object Comparison Tool to uniquely relate a column to a specific table because there is no connection to DB2 at the time the compare process is run (the object definitions also might originate from DDL). This situation occurs when a view references two tables and there is an unqualified reference to a column. DB2 Object Comparison Tool checks if a rename might be the reason for the difference and indicates this in the report. If there are differences, the final outcome is not affected and the view is changed accordingly.

Restriction 2: Rename of an implicit index is not supported.

Restriction 3: Rename of an auxiliary table is not supported.

Table columns

Table columns are matched based on column name. If column positions are different, the table is dropped and re-created to reflect the source sequence of columns.

Column names that are not found in the source file are considered dropped and are removed from the target table unless suppress drop of columns is specified in the Generate Compare Jobs panel, Figure 56 on page 106.

Column names that are not found in the target file are considered new and are added to the target table. If the source and target tables are identical except for one or more appended columns, the target table is altered to add the new columns if the column attributes are acceptable. Otherwise, the table is dropped and re-created.

Related Reading: Refer to the *DB2 Universal Database™ for z/OS SQL Reference* for additional information.

Table drop/re-create without data conversion

Under certain conditions Object Compare can determine that the step that occurs between the unload and load steps to convert the data is not necessary. Performance can improve when the conversion step is omitted from the batch job.

In general, DB2 Object Comparison Tool will not generate a conversion step when the following table modifications are made:

- The table is renamed.
- Columns in the table are:
 - Moved
 - Renamed
 - Deleted
 - Inserted with an attribute of WITH DEFAULT or NULLS
- Only the attributes of the column are changed.

The data types and lengths are changed according to the matrix in the following table:

Table 10. Matrix for data type and length changes that do not require data conversion. The following table describes the matrix used by DB2 Object Comparison Tool to determine whether the data conversion step between unload and load can be skipped.

| From data type | To data type | | | | | | | | | | | | | | | |
|----------------|--------------|-----|-----|-------|------|----------|-----------|------|------|------------|-----|---------|----------------|----------------|---------|-------------|
| | SMALL INT | INT | DEC | FLOAT | CHAR | VAR CHAR | LVAR CHAR | DATE | TIME | TIME STAMP | RID | BIG INT | DEC FLOAT (16) | DEC FLOAT (34) | BIN ARY | VAR BIN ARY |
| SMALLINT | Y | Y | A | Y | - | - | - | - | - | - | - | Y | Y | Y | - | - |
| INT | - | Y | A | Y | - | - | - | - | - | - | - | Y | Y | Y | - | - |
| DEC | A | A | A | Y | - | - | - | - | - | - | - | Y | Y | Y | - | - |
| FLOAT(1-21) | - | - | - | Y | - | - | - | - | - | - | - | Y | Y | Y | - | - |
| FLOAT(22-53) | - | - | - | Y | - | - | - | - | - | - | - | Y | Y | Y | - | - |
| CHAR | A | A | A | - | Y | Y | Y | - | - | - | - | - | - | - | Y | - |
| VARCHAR | A | A | A | - | Y | Y | Y | - | - | - | - | - | - | - | - | Y |
| LVARCHAR | - | - | - | - | Y | Y | Y | - | - | - | - | - | - | - | - | - |
| DATE | - | - | - | - | - | - | - | Y | - | - | - | - | - | - | - | - |
| TIME | - | - | - | - | - | - | - | - | Y | - | - | - | - | - | - | - |
| TIMESTAMP | - | - | - | - | - | - | - | Y | Y | Y | - | - | - | - | - | - |
| RID | - | - | - | - | - | - | - | - | - | - | Y | - | - | - | - | - |
| BIGINT | Y | Y | Y | Y | - | - | - | - | - | - | - | Y | - | - | - | - |
| DECFLOAT(16) | Y | Y | Y | Y | - | - | - | - | - | - | - | - | Y | Y | - | - |

Table 10. Matrix for data type and length changes that do not require data conversion (continued). The following table describes the matrix used by DB2 Object Comparison Tool to determine whether the data conversion step between unload and load can be skipped.

| From data type | To data type | | | | | | | | | | | | | | | |
|----------------|--------------|-----|-----|-------|------|----------|-----------|------|------|------------|-----|---------|----------------|----------------|--------|------------|
| | SMALL INT | INT | DEC | FLOAT | CHAR | VAR CHAR | LVAR CHAR | DATE | TIME | TIME STAMP | RID | BIG INT | DEC FLOAT (16) | DEC FLOAT (34) | BINARY | VAR BINARY |
| DECFLOAT(34) | Y | Y | Y | Y | - | - | - | - | - | - | - | - | Y | Y | - | - |
| BINARY | - | - | - | - | Y | - | - | - | - | - | - | - | - | - | Y | - |
| VARBINARY | - | - | - | - | - | Y | - | - | - | - | - | - | - | - | - | Y |

Notes:

- Y = YES, data conversion is always skipped.
- A = ACTION, data conversion is normally performed, with truncation, if necessary. If the new column can accommodate the data, data conversion is skipped for the following conversion types:
 - smallint to decimal
 - integer to decimal
 - decimal to smallint
 - decimal to integer
 - decimal to decimal

However, if the scale of the decimal type is changed, the data conversion is performed.

When the data conversion step is skipped, a converted unload data set is not created.

Triggers

Triggers are represented as character strings that contain CREATE TRIGGER statements. To apply masks to the trigger definition, triggers are parsed and the language elements are identified. Masks are applied to the source trigger elements where masks are applicable, and the triggers are compared element by element.

The only exception to this process is that to successfully compare an unqualified name to a qualified name, the compare program attempts to determine implicit qualifiers for unqualified names. If the trigger has changed, the change is reported.

The sequence in which triggers are created is important because they are executed in the same sequence by DB2. To maintain the correct sequence, all triggers for a table are processed at the same time.

The manner in which the batch compare program processes triggers depends on the value that you entered in the Suppress DROP of target field in the Generate Compare Jobs panel, Figure 56 on page 106. The following table provides more information.

Table 11. Trigger comparison process. The following table describes how different types of triggers are handled when the "Suppress DROP of target" field is set to No or Yes.

| Suppress DROP of target objects No | Suppress DROP of target objects Yes |
|---|---|
| Source file sequence and contents are used. | Source file sequence and contents are used for all triggers in the source file. |
| Triggers are compared, one by one, based on the trigger name. | Triggers are compared, one by one, based on the trigger name. |
| If a trigger is not in the target file or if the compare finds a difference, the trigger is added or dropped and re-created. All subsequent triggers are dropped and, if applicable, re-created to maintain the correct sequence. | If a trigger is not in the target file or if the compare finds a difference, the trigger is added or dropped and re-created. All subsequent triggers are dropped and, if applicable, re-created to maintain the correct sequence. |

Table 11. *Trigger comparison process (continued)*. The following table describes how different types of triggers are handled when the "Suppress DROP of target" field is set to No or Yes.

| Suppress DROP of target objects No | Suppress DROP of target objects Yes |
|---|--|
| Only triggers found in the target file are dropped. | To avoid violating the sequence of triggers in the source file, only triggers that are found in the target file appear in the first possible position. This approach maintains the original position of these triggers in the target file. |

Views

Views are represented as character strings that contain CREATE VIEW statements. To apply masks to the view definition, views are parsed and the language elements are identified. Masks are applied to the source view elements where masks are applicable, and the views are compared element by element.

The only exception to this process is that to successfully compare an unqualified name to a qualified name, the compare program attempts to determine implicit qualifiers for unqualified names. If the view changes, the change is reported and the view definition, changed or not, is stored.

When the batch compare program has processed all views, it analyzes two types of dependencies:

View dropped

A view is dropped if one of the base tables or views that is referred to was dropped. If a view is dropped, it is re-created regardless of whether it was changed.

View dependent on another view

The sequence in which views are created is important because a view can refer to another view. The stored view definitions are sequenced to take this into account.

This behavior means that CREATE VIEW statements do not necessarily appear in the sequence in which they were processed.

ALIAS for SEQUENCE not supported

DB2 Version 11 supports creating ALIAS for SEQUENCE, however, ALIAS for SEQUENCE is not included in comparisons.

Changing or unloading tables with LOBs

Compare scenarios involving LOB objects now have expanded capabilities to allow changing the objects.

Changes to objects with LOB columns was previously restricted or offered limited capability. Additionally, LOB column data was unloaded to the SYSREC data set, with limitations on the maximum record length allowed.

Objects with LOB columns can now be unloaded with base table data going to the SYSREC data set and the LOB column data going to the data set as directed by the LOB TEMPLATE. This capability requires that the apply job be built as a work

statement list. See the *DB2 Administration Tool User's Guide and Reference for z/OS (SC19-3033-05)* for details on using the utility template to unload data from LOBs and to run a work statement list.

Condition codes

When you run batch compare reports with LOBs, the following condition codes are issued by the GOC2CMP program:

- 0 Ended normally.
- 4 Warning issued. Please review output.
- >4 Error found. Please review output.

LOB restrictions

There are limited LOB column changes for the DB2 Object Comparison Tool. The only possible conversions are:

CHAR, VARCHAR -> CLOB, BLOB
GRAPHIC, VARGRAPHIC -> DBCLOB

The reverse sequence of LOB column to non-LOB column is not supported (for example CLOB to CHAR). Within the comparison report, one of the following messages can appear:

- (E) This type change is not supported.
- (W) This type change is not supported.

LOB column length reduction can cause a failure during compare as this is not supported. Within the comparison report, one of the following messages can appear:

- (E) LOB column length is reduced. This is not supported by DB2 Object Comparison Tool. Manual action is required if you want to reduce the length of a LOB column.
- (W) LOB column length is reduced. This is not supported by DB2 Object Comparison Tool.
- (W) LOB column length will be reduced when recovering the change. Manual action will be required to recover data for this table.

Older version files containing tables with LOBs cannot be processed. The following message appears:

- (E) The version files are generated by a previous version of the product. The version file must be re-created because the internal representation of auxiliary tables in version files has changed.

If the base table containing LOB column(s) is dropped and re-created, the explicit auxiliary table is re-created 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 re-created.

Chapter 8. Creating a Change Management batch job to run compare

The DB2 Admin Change Management batch interface can be used to run DB2 Object Comparison Tool in batch to define or propagate a change that can be managed by DB2 Admin Change Management.

In this mode of the batch interface compare is run to compare a source and a target. The compare report and a delta change file that describes the differences is generated.

When invoking the Change Management batch interface the following parameter must be specified to run compare: **action_compare** = 'Y'

By default the generated delta change file is imported as a new registered change and analyzed on the local system. The importing of the delta change file as a new change can be prevented by specifying the following Change Management batch parameter: **action_import_change** = 'N'

The DB2 Admin Users Guide lists all of the batch interface parameters that can be used to customize the compare run.

Specifying source objects to be compared

About this task

The type of compare source is specified using the **source_type** parameter. Refer to the definition of the **source_type** parameter for the supported compare source types and the other parameters that can be used based on the type of source specified.

Specifying target objects to be compared

About this task

The type of compare target is specified using the **target_type** parameter. Refer to the definition of the **target_type** parameter for the supported compare target types and the other parameters that can be used based on the type of target specified.

Specifying the compare masks

About this task

Using masks for compare is optional.

The masks to use can be specified in the following ways:

- Use the **compare_mask_dsn** parameter to specify the name of an existing data set that contains the compare masks.
- Use the **compare_mask_owner** and **compare_mask_name** parameters to specify an existing mask specification that is in the Change Management database.

- Pre-allocate the compare masks file with DD name of MASKS.

Use the **compare_mask_dsn** parameter or use of the **compare_mask_owner** and **compare_mask_name** parameters override the use of any pre-allocated compare masks file (MASKS DD).

Also, you cannot specify both the use of the **compare_mask_dsn** parameter and the use of **compare_mask_owner** and **compare_mask_name** parameters.

Specifying compare ignore fields

About this task

Using ignore fields for compare is optional. The ignore fields to use can be specified in the following ways:

1.
 - a. Use the **compare_ignore_fields_dsn** parameter to specify the name of an existing data set that contains the compare ignore fields.
 - b. Use the **compare_ignore_fields_owner** and **compare_ignore_fields_name** parameters to specify an existing ignore fields specification that is in the Change Management database.
 - c. Pre-allocate compare ignore fields file with DD name of IGNORES.

If method (a) or (b) is specified it overrides the use of any pre-allocated compare ignore fields file (IGNORES DD). Method (a) and (b) cannot both be specified.

Specifying compare ignore changes

About this task

Using ignore changes is optional. The **compare_ignore_changes_owner** and **compare_ignore_changes_name** parameters can be used to specify an existing ignore changes specification that is stored in the Change Management database.

Specifying compare exclude

About this task

Using exclude parameter is optional. You specify an existing exclude specification that is stored in the Change Management database. For compare source, use the parameters **source_exclude_owner** and **source_exclude_name** parameters. For compare target, use the parameters **target_exclude_owner** and **target_exclude_name** parameters.

Chapter 9. DB2 Object Comparison Tool condition codes

These topics provide information about the condition codes DB2 Object Comparison Tool issues.

The following condition codes are issued by the ADB2GEN program. ADB2GEN is used to create a version file.

If you receive a condition code greater than zero, review the messages carefully.

0 The run was successful.

4

- A parameter error occurred. The parameter was ignored or the default was used. No generate requests were issued.
- Requested object was not found. A warning is issued.

8

- No parameters were found. Processing ended.
- The DB2 version is not supported. Other errors might be issued.

12

- The DB2 version is not supported. Processing ended.
- The remote location is not defined or is not a DB2 z/OS system. This is an internal error or limitation.
- Other severe errors were detected.

16 A severe error occurred.

The following condition codes are issued by the GOC2CMP program.

0 The GOC2CMP was run successfully.

4 Index not dropped which can lead to loss of referential integrity. Refer to the listed error message.

6 SQL PL functions have been bypassed because the BYPASSSQLPL parameter was specified. Examine the generated APPLY job or work statement list to verify that the content is complete.

8 There are problems with referential constraints. Manual action is required. Refer to the listed error message.

16 A severe error occurred. Refer to the listed error message.

In addition to the condition codes for GOC2CMP, GOC2DTC can issue the following condition code.

12 A quoted string is too long.

If the messages that are associated with these codes do not contain adequate information regarding the actions you should take, use the information in the following table to diagnose common problems before you contact IBM Software Support.

Before contacting IBM Support

If you receive DB2 Object Comparison Tool error messages that do not contain adequate information regarding the actions you should take, use the information contained in this section to diagnose common problems before you contact the IBM Software Support. The information that you gather to diagnose the problem is required when you open an incident with the DB2 Object Comparison Tool Support team.

The following table provides the documentation that is required for errors that can occur during the compare process.

Table 12. Documentation required by IBM Software Support

| Step or function | Documentation required |
|---|--|
| All failures | Always provide the complete job output |
| Generate version file from DDL (Step <i>fmDDL</i> - program GOC2DTC) | The DDL file |
| Extract version file from the catalog (Step <i>fmDB2</i> - program ADB2GEN) | <ul style="list-style-type: none">• The original DDL as defined in the catalog, or• An extract of the source and target objects selected using GEN |
| Compare (Step T03COMP - program GOC2CMP) | <ul style="list-style-type: none">• Source and target version files• Source and target DDL as defined in the catalog• MASK and IGNORES input files• CHANGES file output, if any• IFF file output, if any |
| Apply (multiple steps and programs) | <ul style="list-style-type: none">• The work statement list, if applicable• The JCL that was submitted• If the error is a logic error in DDL or utility statements or if you are unsure, provide the utility control statements. These might be included in the complete job output. |

Chapter 10. Batch compare report format

The batch compare program produces a report that contains data based on the change reporting options selected on the Generate Compare Jobs panel.

You can specify the specific information included in the report on the Specify Compare Reporting Options panel:

```

Compare ----- Generate Compare Jobs -----
Option ==>

Specify the following for DB2 Object Comparison Tool:

Worklist information:
  Worklist name . . . . . : PQ76055N (also used as middle qualifier in DSNs)
-----
Co  Compare ----- Specify Compare Reporting Options ----- 12:18
-----
    Report options for Compare:
      Only changed objects . . : Yes      (Yes/No)
    Ignore fields:
Ch   User specified . . . . . : Yes      (Yes/No)
     System generated . . . . . : Yes      (Yes/No)
Da  Translation masks . . . . . : Yes      (Yes/No)
     Summary report . . . . . : Yes      (Yes/No)
     Object count report . . . . : Yes      (Yes/No)
     Conversion report . . . . . : YES      (Yes/No)
Op  -----
  
```

Figure 94. Generate Compare Jobs pop-up panel 1 (GOC5R0)

In addition, you can specify the report format on the Change Options Common to Change Functions panel (ADB2PCO).

- If you specify T for processing order, the batch report will display results for objects sorted by type. For example, a report generated from the T processing order might display all databases, followed by all table spaces, followed by all tables.
- If you specify H (default) for processing order, the batch report will display results for all the object types grouped by database. In this hierarchical format, each database will be followed by objects in that database. For example, table spaces in a database will follow the database, tables in a table space will follow the table space, and indexes over each table will follow the table.

Note: There are exceptions to this order. Temporal tables and history tables follow after all databases because they need to be processed after all table spaces are processed.

Four sample reports are shown in the following topics. The sample reports contain the following information:

Only changed objects

If you specify No for this option, the program produces a complete object comparison report containing all objects. If you specify Yes for this option, the report only contains the changed, deleted, dropped, and dropped/re-created objects.

The source and target version files are described at the top of the object comparison report. You can add free-form text when each version file is created through the ISPF full screen interface. This text is written to the report and is followed by:

- Input for the source and target for the extraction of the source and target objects.

If the objects were extracted from a DB2 catalog, this value is the DB2 subsystem ID. If the objects were extracted from a file with saved DDL statements, DDL* is indicated.

- When the extract was performed.
- By whom the extract was performed.

UNKNOWN is shown if the user ID is not known.

If long names are used (to exploit DB2 Version 8 function mode), authorization IDs or names can span lines. Object Comparison Tool will try not to split an authorization ID or a name if possible.

Information about the comparison comprises the remainder of this section. In the Compare DB2 objects sample report 2, for example, the first comparison involves source object RRR8D81A.DSN8S81D and target object TTT8D81A.DSN8S81D (a typical example of a comparison between a development and production system). The results of the comparison can include:

- Added objects

Added objects are reported as objects not found in the target. In this case, objects are added. For example, the report shows that a new view named VNDR230.VDEPTS is to be added.

- Dropped objects

Dropped objects are reported as objects not found in the source. In this case, objects are dropped. For example, the report shows that a new view named VNDR230.VASTRDE1 is to be dropped.

- Compared objects

For compare objects, a sequence of information is reported:

- Object identification

The object type and object names of source and target objects are listed. For example, the Compare DB2 objects sample report 2 shows that table space source RRR8D81A.DSN8S81D and target TTT8D81A.DSN8S81D are being compared. The object names might be different, as in this case, because they are shown with no masks applied.

- Differences

If differences are found, they are reported one by one. The report indicates how the upgrade will be performed:

- (A) means that ALTER object can be used.
- (D) means that the object will need to be dropped and re-created.

- Summary

This summarizes the action that will be taken to upgrade the object or an indication that no change to the object was detected.

Other messages that might be reported include the following:

- (E) Error message
- (W) Warning message
- (I) Informational message

Messages can contain return codes, which provide additional context based on your situation.

Note: The comparison process only writes a file of the changes that are necessary to upgrade target objects to match source objects. No actual changes are made. For information about implementing the changes, see “Running a work statement list” on page 136.

Ignore fields

This provides a listing of the ignore fields used. You can specify Yes or No in the user-specified and system generated fields to indicate if you want these types of ignore fields included in the report.

Translation masks

This provides a listing of the translation masks used. Specify Yes or No to indicate if you want the report to include the translation masks used.

Summary report

This provides a summary report that contains one line for each object that was compared and the result of the comparison. Specify Yes or No to indicate if you want to produce a Summary report. If long names are used (to exploit DB2 Version 8 function mode), authorization IDs or names cannot be shown on a single line. In this case, the names are truncated. For long authorization IDs, the first eight characters are shown, followed by a > to indicate a long authorization ID. If an object name is long, the first 18 characters are shown, followed by a > to indicate a long name.

Object count report

This provides a report that shows how many objects were compared per object type. Specify Yes or No to indicate if you want the report to include an Object count report.

The following different sample reports for comparing DB2 objects are provided:

- The Sample Batch Compare Report shown in the Compare DB2 objects sample report 1 shows all fields that were ignored.
- The Sample Batch Compare Report shown in Compare DB2 objects sample report 2 shows user-specified ignores and contains a section that provides a full comparison report.
- The Sample Batch Compare Report shown in Compare DB2 objects sample report 3 shows a subset of the report that shows where additional masks were specified. This report also contains a section that shows system-generated ignore fields and a section that shows only the objects that were changed, deleted, or dropped.
- The Sample Batch Compare Report shown in Compare DB2 objects sample report 4 shows a subset of the compare report in which LOB objects are converted from explicit to implicit.
- The Sample Batch Compare Report shown in Compare DB2 objects sample report 5 shows a subset of the compare report for which the comparison was run for one object type, rather than all object types.
- The sample summary conversion report shows truncations and conversions that will take place when the change runs.

- The list of possible conversion errors shows possible conversion errors that might occur due to truncation during conversion of data types.

The following topics provide additional information:

- “Compare DB2 Objects sample report 1”
- “Compare DB2 Objects sample report 2” on page 177
- “Compare DB2 Objects sample report 3” on page 179
- “Compare DB2 Objects sample summary conversion report” on page 184
- “Possible conversion errors” on page 184

Related concepts:

Chapter 7, “Batch compare program,” on page 157

The batch compare program is run when you specify options on the Generate Compare Jobs panel and generate a compare batch job. This program compares two sets of DB2 objects, reports all differences, and writes all changes to a file. This file is used to generate updates to upgrade target objects to the level of source objects.

Related reference:

“Generate compare job options” on page 113

Specify the criteria that you want to use to generate the work statement list or JCL for your compare job.

Compare DB2 Objects sample report 1

The sample batch compare report shown in the following figure contains all sections of the batch compare report. The FIELDS IGNORED WHEN COMPARING SOURCE AND TARGET OBJECTS section shows all fields that were ignored. The OBJECT COMPARISON REPORT section shows only the objects that were changed, added, or dropped.

```
-----
GOC2CMP - Compare DB2 Objects                                     2006-06-09 19:01
-----

Database 2 Object Comparison Tool
5697-L40 (C) Copyright IBM Corporation 2001, 2006.
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.

Parameters for this run:

Suppress DROP of objects : No
Suppress DROP of columns : No
Suppress adding columns : No

TRANSLATION MASKS
=====

DBNAME      : RRR8D81A                                     , TTT8D81A

FIELDS IGNORED WHEN COMPARING SOURCE AND TARGET OBJECTS
=====

SYSAXRELS   : AUXRELOBID(S), IBMREQD(S)
SYSCHECKS   : DBID(S), OBID(S), TIMESTAMP(S), RBA(S), IBMREQD(S)
SYSCOLAUTH   : TIMESTAMP(S), DATEGRANTED(S), TIMEGRANTED(S), IBMREQD(S),
              CONTOKEN(S), GRANTEDTS(S)
```

SYSCOLUMNS : COLCARD(S), HIGH2KEY(S), LOW2KEY(S), IBMREQD(S), STATSTIME(S),
 COLCARDF(S), CREATEDTS(S), ALTEREDTS(S)
 SYSDATABASE : DBID(S), IBMREQD(S), CREATEDBY(S), TIMESTAMP(S), CREATEDTS(S),
 ALTEREDTS(S), BPOOL(U), INDEXBP(U)
 SYSDATATYPES : CREATEDBY(S), DATATYPEID(S), CREATEDTS(S), IBMREQD(S)
 SYSDBAUTH : TIMESTAMP(S), DATEGRANTED(S), TIMEGRANTED(S), GRANTEETYPE(S),
 IBMREQD(S), GRANTEDTS(S)
 SYSFIELDS : IBMREQD(S)
 SYSFOREIGNKEYS : IBMREQD(S)
 SYSINDEXES : CLUSTERED(S), DBID(S), OBID(S), ISOBID(S), INDEXSPACE(S),
 FIRSTKEYCARD(S), FULLKEYCARD(S), NLEAF(S), NLEVELS(S),
 SPACE(S), IBMREQD(S), CLUSTERRATIO(S), CREATEDBY(S),
 IOFACTOR(S), PREFETCHFACTOR(S), STATSTIME(S),
 FIRSTKEYCARDF(S), FULLKEYCARDF(S), CREATEDTS(S), ALTEREDTS(S),
 COPYLRN(S), CLUSTERRATIOF(S), SPACEF(S), BPOOL(U)
 SYSINDEXPART : CARD(S), FAROFFPOS(S), LEAFDIST(S), NEAROFFPOS(S), IBMREQD(S),
 SPACE(S), STATSTIME(S), FAROFFPOSF(S), NEAROFFPOSF(S),
 CARDF(S), ALTEREDTS(S), SPACEF(S), DSNUM(S), EXTENTS(S),
 PSEUDO_DEL_ENTRIES(S), LEAFNEAR(S), LEAFFAR(S), CREATEDTS(S),
 PQTY(U), SQTY(U), STORTYPE(U), STORNAME(U), VCATNAME(U),
 FREEPAGE(U), PCTFREE(U), SECQTYI(U)
 SYSKEYCOLUSE : IBMREQD(S)
 SYSKEYS : IBMREQD(S)
 SYSPACKDEP : IBMREQD(S)
 SYSPLANDEP : IBMREQD(S)
 SYSPARMS : ROUTINEID(S), DATATYPEID(S), CAST_FUNCTION_ID(S), IBMREQD(S)
 SYSRELS : IBMREQD(S), RELOBID1(S), RELOBID2(S), TIME_STAMP(S)
 SYSRESAUTH : TIMESTAMP(S), DATEGRANTED(S), TIMEGRANTED(S), IBMREQD(S),
 GRANTEDTS(S)
 SYSROUTINEAUTH : GRANTEDTS(S), IBMREQD(S)
 SYSROUTINES : CREATEDBY(S), ROUTINEID(S), CREATEDTS(S), ALTEREDTS(S),
 IBMREQD(S), PARM1(S), PARM2(S), PARM3(S), PARM4(S), PARM5(S),
 PARM6(S), PARM7(S), PARM8(S), PARM9(S), PARM10(S), PARM11(S),
 PARM12(S), PARM13(S), PARM14(S), PARM15(S), PARM16(S),
 PARM17(S), PARM18(S), PARM19(S), PARM20(S), PARM21(S),
 PARM22(S), PARM23(S), PARM24(S), PARM25(S), PARM26(S),
 PARM27(S), PARM28(S), PARM29(S), PARM30(S)
 SYSSCHEMAAUTH : GRANTEDTS(S), IBMREQD(S)
 SYSSEQUENCES : NAME(S), SEQUENCEID(S), CREATEDBY(S), CREATEDTS(S),
 ALTEREDTS(S), IBMREQD(S)
 SYSSEQUENCEAUTH : CONTOKEN(S), GRANTEDTS(S), IBMREQD(S)
 SYSSEQUENCESDEP : BSEQUENCEID(S), IBMREQD(S)
 SYSSTOGROUP : VPASSWORD(S), SPACE(S), SPCDATE(S), IBMREQD(S), CREATEDBY(S),
 STATSTIME(S), CREATEDTS(S), ALTEREDTS(S), SPACEF(S)
 SYSSYNONYMS : IBMREQD(S), CREATEDBY(S), CREATEDTS(S)
 SYSTABAUTH : TIMESTAMP(S), DATEGRANTED(S), TIMEGRANTED(S), IBMREQD(S),
 GRANTEDTS(S)
 SYSTABCONST : CREATEDTS(S), IBMREQD(S)
 SYSTABLEPART : CARD(S), FARINDREF(S), NEARINDREF(S), PERCACTIVE(S),
 PERCDROP(S), IBMREQD(S), CHECKRID(S), SPACE(S), PAGESAVE(S),
 STATSTIME(S), CHECKRID5B(S), EPOCH(S), CARDF(S), ALTEREDTS(S),
 SPACEF(S), DSNUM(S), EXTENTS(S), LIMITKEY_INTERNAL(S),
 CREATEDTS(S)
 SYSTABLES : DBID(S), OBID(S), CLUSTERRID(S), CARD(S), NPAGES(S),
 PCTPAGES(S), IBMREQD(S), PARENTS(S), CHILDREN(S), KEYOBID(S),
 CHECKRID(S), CREATEDBY(U), CREATEDTS(S), ALTEREDTS(S),
 RBA1(S), RBA2(S), PCTROWCOMP(S), STATSTIME(S), CARDF(S),
 CHECKRID5B(S), NPAGESF(S), SPACEF(S), AVGWLEN(S),
 RELCREATED(S)
 SYSTABLESPACE : DBID(S), OBID(S), PSID(S), NTABLES(S), NACTIVE(S), SPACE(S),
 IBMREQD(S), ROOTNAME(S), ROOTCREATOR(U), CREATEDBY(S),
 STATSTIME(S), CREATEDTS(S), ALTEREDTS(S), NACTIVEF(S),
 SPACEF(S), BPOOL(U)
 SYSTRIGGERS : DBID(S), OBID(S), CREATEDBY(S), CREATEDTS(S), IBMREQD(S)
 SYSVIEWDEP : IBMREQD(S)
 SYSVIEWS : IBMREQD(S), RELCREATED(S), REFRESH_TIME(S), SIGNATURE(S)
 SVOLUMES : IBMREQD(S)

- (S) System ignore. Set automatically by compare
Also set for fields only used by newer versions of DB2
- (U) User ignore. Requested by user input
(U) is reported for fields that are both System and User ignores

GOC2CMP - Compare DB2 Objects

2006-06-09 19:01

OBJECT COMPARISON REPORT
=====

Only changed, added and deleted objects will be reported

Source: VIEW ADDED
Extracted from location *FROM DDL FILE* at 2006-06-09 18:57 by UNKNOWN

Target: TABLESPACE TTT8S81D FROM CAT
Extracted from DSN8 at 2006-06-09 18:57 by VNDR230

Target system is DB2 Release 810

View VNDR230.VDEPTS not found on target
New View VNDR230.VDEPTS will be added
Authorisations for View VNDR230.VDEPTS will be copied from source

COMPARISON SUMMARY REPORT
=====

| Obtyp | Source Object | Target Object | Result | Object type |
|-------|-------------------|-------------------|-----------|-------------|
| S | RRR8D81A.DSN8S81D | TTT8D81A.DSN8S81D | No change | Tablespace |
| T | VNDR230.DEPT | VNDR230.DEPT | No change | Table |
| X | VNDR230.XDEPT1 | VNDR230.XDEPT1 | No change | Index |
| X | VNDR230.XDEPT2 | VNDR230.XDEPT2 | No change | Index |
| X | VNDR230.XDEPT3 | VNDR230.XDEPT3 | No change | Index |
| R | RDD | RDD | No change | Relation |
| R | RDE | RDE | No change | Relation |
| V | VNDR230.VASTRDE1 | VNDR230.VASTRDE1 | No change | View |
| V | VNDR230.VASTRDE2 | VNDR230.VASTRDE2 | No change | View |
| V | VNDR230.VDEPMG1 | VNDR230.VDEPMG1 | No change | View |
| V | VNDR230.VDEPT | VNDR230.VDEPT | No change | View |
| V | VNDR230.VDEPTS | | Added | View |
| V | VNDR230.VEMPDPT1 | VNDR230.VEMPDPT1 | No change | View |
| V | VNDR230.VHDEPT | VNDR230.VHDEPT | No change | View |
| V | VNDR230.VPHONE | VNDR230.VPHONE | No change | View |

COMPARISON COUNTS REPORT
=====

| Object type | On source | On target | Compared | Added | Dropped | Altered | Not Added |
|-------------|-----------|-----------|----------|-------|---------|---------|-----------|
| Tablespaces | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| Tables | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| Indexes | 3 | 3 | 3 | 0 | 0 | 0 | 0 |
| Views | 8 | 7 | 7 | 1 | 0 | 0 | 0 |
| Relations | 2 | 2 | 2 | 0 | 0 | 0 | 0 |

Compare DB2 Objects sample report 2

The sample batch compare report shown in the following figure contains all sections of the batch compare report. The **FIELDS IGNORED WHEN COMPARING SOURCE AND TARGET OBJECTS** section shows only user-specified ignores. The **OBJECT COMPARISON REPORT** section is a full comparison report.

GOC2CMP - Compare DB2 Objects 2006-06-10 09:20

Database 2 Object Comparison Tool
5697-L40 (C) Copyright IBM Corporation 2001, 2006.
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.

TRANSLATION MASKS

DBNAME : RRR8D81A , TTT8D81A

FIELDS IGNORED WHEN COMPARING SOURCE AND TARGET OBJECTS

Only user requested ignore fields are reported

SYSINDEXES : BPOOL(U)
SYSINDEXPART : PQTY(U), SQTY(U), STORTYPE(U), STORNAME(U), VCATNAME(U),
FREEPAGE(U), PCTFREE(U), SECQTYI(U)
SYSTABLEPART : PQTY(U), SQTY(U), STORTYPE(U), STORNAME(U), VCATNAME(U),
FREEPAGE(U), PCTFREE(U), SECQTYI(U)
SYSTABLESPACE : BPOOL(U), MAXROWS(U)

GOC2CMP - Compare DB2 Objects 2006-06-10 09:20

OBJECT COMPARISON REPORT

Source: VIEW ADDED, FULL REPORT
Extracted from location *FROM DDL FILE* at 2006-06-10 09:16 by UNKNOWN

Target: TABLESPACE TTT8S81D FROM CATLG
Extracted from DSN8 at 2006-06-10 09:16 by VNDR230

Target system is DB2 Release 810

Compare tablespace source(RRR8D81A.DSN8S81D) and target(TTT8D81A.DSN8S81D)
No changes to Tablespace

Grant(target): Grantor=VNDR230 Grantee:PUBLIC (Kept)

Compare table source(VNDR230.DEPT) and target(VNDR230.DEPT)

No changes to Table
Grant(target): Grantor=VNDR230 Grantee:PUBLIC* (Kept)

Compare index source(VNDR230.XDEPT1) and target(VNDR230.XDEPT1)

No changes to Index

Compare index source(VNDR230.XDEPT2) and target(VNDR230.XDEPT2)

No changes to Index

Compare index source(VNDR230.XDEPT3) and target(VNDR230.XDEPT3)

No changes to Index

View VNDR230.VASTRDE1 not found on source
View VNDR230.VASTRDE1 will be dropped

View VNDR230.VASTRDE2 not found on source
View VNDR230.VASTRDE2 will be dropped

View VNDR230.VDEPMG1 not found on source
View VNDR230.VDEPMG1 will be dropped

Compare View source(VNDR230.VDEPT) and target(VNDR230.VDEPT)
No changes to View
Grant(target): Grantor=VNDR230 Grantee:PUBLIC* (Kept)

View VNDR230.VDEPTS not found on target
New View VNDR230.VDEPTS will be added
Authorisations for View VNDR230.VDEPTS will be copied from source

View VNDR230.VEMPDPT1 not found on source
View VNDR230.VEMPDPT1 will be dropped

Compare View source(VNDR230.VHDEPT) and target(VNDR230.VHDEPT)
No changes to View
Grant(target): Grantor=VNDR230 Grantee:PUBLIC* (Kept)

View VNDR230.VPHONE not found on source
View VNDR230.VPHONE will be dropped

Compare Referential Constraint source(RDD) and target(RDD)
No changes to Referential constraint

Compare Referential Constraint source(RDE) and target(RDE)
No changes to Referential constraint

COMPARISON SUMMARY REPORT
=====

| Obytp | Source Object | Target Object | Result | Object type |
|-------|-------------------|-------------------|-----------|-------------|
| S | RRR8D81A.DSN8S81D | TTT8D81A.DSN8S81D | No change | Tablespace |
| T | VNDR230.DEPT | VNDR230.DEPT | No change | Table |
| X | VNDR230.XDEPT1 | VNDR230.XDEPT1 | No change | Index |
| X | VNDR230.XDEPT2 | VNDR230.XDEPT2 | No change | Index |
| X | VNDR230.XDEPT3 | VNDR230.XDEPT3 | No change | Index |
| R | RDD | RDD | No change | Relation |
| R | RDE | RDE | No change | Relation |
| V | | VNDR230.VASTRDE1 | Dropped | View |
| V | | VNDR230.VASTRDE2 | Dropped | View |
| V | | VNDR230.VDEPMG1 | Dropped | View |
| V | VNDR230.VDEPT | VNDR230.VDEPT | No change | View |
| V | VNDR230.VDEPTS | | Added | View |
| V | | VNDR230.VEMPDPT1 | Dropped | View |
| V | VNDR230.VHDEPT | VNDR230.VHDEPT | No change | View |
| V | | VNDR230.VPHONE | Dropped | View |

COMPARISON COUNTS REPORT
=====

| Object type | On source | On target | Compared | Added | Dropped | Altered | Not Added |
|-------------------|-----------|-----------|----------|-------|---------|---------|-----------|
| Schemas | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| User Def Types | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Sequences | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Databases | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Tablespaces | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| Tables | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| Indexes | 3 | 3 | 3 | 0 | 0 | 0 | 0 |
| Aliases | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Storage groups | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Synonyms | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Functions | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Stored procedures | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Triggers | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Views | 3 | 7 | 2 | 1 | 5 | 0 | 0 |
| Relations | 2 | 2 | 2 | 0 | 0 | 0 | 0 |

Compare DB2 Objects sample report 3

The sample batch compare report in the following figure shows a subset of the compare report in which additional masks were specified. The **FIELDS IGNORED WHEN COMPARING SOURCE AND TARGET OBJECTS** section shows only system-generated ignore fields. The **OBJECT COMPARISON REPORT** section shows only the objects that were changed, added, or dropped.

GOC2CMP - Compare DB2 Objects

2006-06-10 10:00

Database 2 Object Comparison Tool
5697-L40 (C) Copyright IBM Corporation 2001, 2006.
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.

TRANSLATION MASKS

=====

| | | | | |
|----------|---|----------|---|----------|
| OWNER | : | AAA* | , | BBB* |
| OWNER | : | TESTSYS | , | PRODOWN |
| GRANTEE | : | TESTX | , | PRODOWN |
| AUTHID | : | VNDOJK2 | , | VNDR230 |
| TBNAME | : | TAB1* | , | XXTAB* |
| NAME | : | VNDOJK2 | , | VNDR230 |
| DBNAME | : | DB01 | , | PRODDB |
| DBNAME | : | RRR8D81A | , | TTT8D81A |
| SGNAME | : | TESTG | , | PRODG |
| BPNAME | : | BP1 | , | BP4 |
| TSBPNAME | : | BP0 | , | BP1 |
| IXBPNAME | : | BP0 | , | BP2 |

Processed top down. First mask that fits a name of a given type will be used

BPNAME will cover TSBPNAME and IXBPNAME
SGNAME will cover TSSGNAME and IXSGNAME
NAME will cover all NAME types except COLNAME
AUTHID will cover SQLID, OWNER, SCHEMA and GRANTOR/GRANTEE

FIELDS IGNORED WHEN COMPARING SOURCE AND TARGET OBJECTS

=====

Only system generated ignore fields are reported
System ignore fields also found in the user ignore input will not be reported

| | | |
|-----------------|---|---|
| SYS_AUXRELS | : | AUXRELOBID(S), IBMREQD(S) |
| SYS_CHECKS | : | DBID(S), OBID(S), TIMESTAMP(S), RBA(S), IBMREQD(S) |
| SYS_COLLAUTH | : | TIMESTAMP(S), DATEGRANTED(S), TIMEGRANTED(S), IBMREQD(S), CONTOKEN(S), GRANTEDTS(S) |
| SYS_COLUMNS | : | COLCARD(S), HIGH2KEY(S), LOW2KEY(S), IBMREQD(S), STATSTIME(S), COLCARDF(S), CREATEDTS(S), ALTEREDTS(S) |
| SYS_DATABASE | : | DBID(S), IBMREQD(S), CREATEDBY(S), TIMESTAMP(S), CREATEDTS(S), ALTEREDTS(S) |
| SYS_DATATYPES | : | CREATEDBY(S), DATATYPEID(S), CREATEDTS(S), IBMREQD(S) |
| SYS_DBAUTH | : | TIMESTAMP(S), DATEGRANTED(S), TIMEGRANTED(S), GRANTEETYPE(S), IBMREQD(S), GRANTEDTS(S) |
| SYS_FIELDS | : | IBMREQD(S) |
| SYS_FOREIGNKEYS | : | IBMREQD(S) |
| SYS_INDEXES | : | CLUSTERED(S), DBID(S), OBID(S), ISOBID(S), INDEXSPACE(S), |

```

FIRSTKEYCARD(S), FULLKEYCARD(S), NLEAF(S), NLEVELS(S),
SPACE(S), IBMREQD(S), CLUSTERRATIO(S), CREATEDBY(S),
IOFACTOR(S), PREFETCHFACTOR(S), STATSTIME(S),
FIRSTKEYCARDF(S), FULLKEYCARDF(S), CREATEDTS(S), ALTEREDTS(S),
COPYLRN(S), CLUSTERRATIOF(S), SPACEF(S)
SYSINDEXPART : CARD(S), FAROFFPOS(S), LEAFDIST(S), NEAROFFPOS(S), IBMREQD(S),
SPACE(S), STATSTIME(S), FAROFFPOSF(S), NEAROFFPOSF(S),
CARDF(S), ALTEREDTS(S), SPACEF(S), DSNUM(S), EXTENTS(S),
PSEUDO_DEL_ENTRIES(S), LEAFNEAR(S), LEAFFAR(S), CREATEDTS(S)
SYSKEYCOLUSE : IBMREQD(S)
SYSKEYS : IBMREQD(S)
SYSPACKDEP : IBMREQD(S)
SYSPLANDEP : IBMREQD(S)
SYSPARMS : ROUTINEID(S), DATATYPEID(S), CAST_FUNCTION_ID(S), IBMREQD(S)
SYSRELS : IBMREQD(S), RELOBID1(S), RELOBID2(S), TIMESTAMP(S)
SYSRESAUTH : TIMESTAMP(S), DATEGRANTED(S), TIMEGRANTED(S), IBMREQD(S),
GRANTEDTS(S)
SYSROUTINEAUTH : GRANTEDTS(S), IBMREQD(S)
SYSROUTINES : CREATEDBY(S), ROUTINEID(S), CREATEDTS(S), ALTEREDTS(S),
IBMREQD(S), PARM1(S), PARM2(S), PARM3(S), PARM4(S), PARM5(S),
PARM6(S), PARM7(S), PARM8(S), PARM9(S), PARM10(S), PARM11(S),
PARM12(S), PARM13(S), PARM14(S), PARM15(S), PARM16(S),
PARM17(S), PARM18(S), PARM19(S), PARM20(S), PARM21(S),
PARM22(S), PARM23(S), PARM24(S), PARM25(S), PARM26(S),
PARM27(S), PARM28(S), PARM29(S), PARM30(S)
SYSSCHEMAAUTH : GRANTEDTS(S), IBMREQD(S)
SYSSEQUENCES : NAME(S), SEQUENCEID(S), CREATEDBY(S), CREATEDTS(S),
ALTEREDTS(S), IBMREQD(S)
SYSSEQUENCEAUTH : CONTOKEN(S), GRANTEDTS(S), IBMREQD(S)
SYSSEQUENCESDEP : BSEQUENCEID(S), IBMREQD(S)
SYSSTOGROUP : VPASSWORD(S), SPACE(S), SPCDATE(S), IBMREQD(S), CREATEDBY(S),
STATSTIME(S), CREATEDTS(S), ALTEREDTS(S), SPACEF(S)
SYSSYNONYMS : IBMREQD(S), CREATEDBY(S), CREATEDTS(S)
SYSTABAUTH : TIMESTAMP(S), DATEGRANTED(S), TIMEGRANTED(S), IBMREQD(S),
GRANTEDTS(S)
SYSTABCONST : CREATEDTS(S), IBMREQD(S)
SYSTABLEPART : CARD(S), FARINDREF(S), NEARINDREF(S), PERCACTIVE(S),
PERCDROP(S), IBMREQD(S), CHECKRID(S), SPACE(S), PAGESAVE(S),
STATSTIME(S), CHECKRID5B(S), EPOCH(S), CARDF(S), ALTEREDTS(S),
SPACEF(S), DSNUM(S), EXTENTS(S), LIMITKEY_INTERNAL(S),
CREATEDTS(S)
SYSTABLES : DBID(S), OBID(S), CLUSTERRID(S), CARD(S), NPAGES(S),
PCTPAGES(S), IBMREQD(S), PARENTS(S), CHILDREN(S), KEYOBID(S),
CHECKRID(S), CREATEDBY(S), CREATEDTS(S), ALTEREDTS(S),
RBA1(S), RBA2(S), PCTROWCOMP(S), STATSTIME(S), CARDF(S),
CHECKRID5B(S), NPAGESF(S), SPACEF(S), AVGWLEN(S),
RELCREATED(S)
SYSTABLESPACE : DBID(S), OBID(S), PSID(S), NTABLES(S), NACTIVE(S), SPACE(S),
IBMREQD(S), ROOTNAME(S), ROOTCREATOR(S), CREATEDBY(S),
STATSTIME(S), CREATEDTS(S), ALTEREDTS(S), NACTIVEF(S),
SPACEF(S)
SYSTRIGGERS : DBID(S), OBID(S), CREATEDBY(S), CREATEDTS(S), IBMREQD(S)
SYSVIEWDEP : IBMREQD(S)
SYSVIEWS : IBMREQD(S), RELCREATED(S), REFRESH_TIME(S), SIGNATURE(S)
SYSVOLUMES : IBMREQD(S)

```

GOC2CMP - Compare DB2 Objects

2006-06-10 10:00

OBJECT COMPARISON REPORT

=====

Only changed, added and deleted objects will be reported

Source: VIEW ADDED, CHANGED ONLY REPT
 Extracted from location *FROM DDL FILE* at 2006-06-10 09:56 by UNKNOWN

Target: TABLESPACE TTT8S81D FROM CATLG
 Extracted from DSN8 at 2006-06-10 09:56 by VNDR230

Target system is DB2 Release 810

View VNDR230.VDEPTS not found on target
 New View VNDR230.VDEPTS will be added
 Authorisations for View VNDR230.VDEPTS will be copied from source

COMPARISON SUMMARY REPORT
 =====

| Obtyp | Source Object | Target Object | Result | Object type |
|-------|-------------------|-------------------|-----------|-------------|
| S | RRR8D81A.DSN8S81D | TTT8D81A.DSN8S81D | No change | Tablespace |
| T | VNDR230.DEPT | VNDR230.DEPT | No change | Table |
| X | VNDR230.XDEPT1 | VNDR230.XDEPT1 | No change | Index |
| X | VNDR230.XDEPT2 | VNDR230.XDEPT2 | No change | Index |
| X | VNDR230.XDEPT3 | VNDR230.XDEPT3 | No change | Index |
| R | RDD | RDD | No change | Relation |
| R | RDE | RDE | No change | Relation |
| V | VNDR230.VASTRDE1 | VNDR230.VASTRDE1 | No change | View |
| V | VNDR230.VASTRDE2 | VNDR230.VASTRDE2 | No change | View |
| V | VNDR230.VDEPMG1 | VNDR230.VDEPMG1 | No change | View |
| V | VNDR230.VDEPT | VNDR230.VDEPT | No change | View |
| V | VNDR230.VDEPTS | | Added | View |
| V | VNDR230.VEMPDPT1 | VNDR230.VEMPDPT1 | No change | View |
| V | VNDR230.VHDEPT | VNDR230.VHDEPT | No change | View |
| V | VNDR230.VPHONE | VNDR230.VPHONE | No change | View |

COMPARISON COUNTS REPORT
 =====

| Object type | On source | On target | Compared | Added | Dropped | Altered | Not Added |
|-------------|-----------|-----------|----------|-------|---------|---------|-----------|
| Tablespaces | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| Tables | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| Indexes | 3 | 3 | 3 | 0 | 0 | 0 | 0 |
| Views | 8 | 7 | 7 | 1 | 0 | 0 | 0 |
| Relations | 2 | 2 | 2 | 0 | 0 | 0 | 0 |

Compare DB2 Objects sample report 4

The sample batch compare report in the following figure shows a subset of the compare report in which LOB objects are converted from explicit to implicit. The OBJECT COMPARISON REPORT includes messages about the converted objects. The summary section of the report lists all objects and the end result of the action performed.

OBJECT COMPARISON REPORT
 =====

Tablespace DB33971.TL971APN not found on target
 New LOB Tablespace DB33971.TL971APN will be added

Compare tablespace source(DB33971.TL971AP1) and target(DB33971.TL971AP1)
No changes to Tablespace

Compare tablespace source(DB33971.TL971AP2) and target(DB33971.TL971AP2)
No changes to Tablespace

Compare tablespace source(DB33971.TL971AP3) and target(DB33971.TL971AP3)
No changes to Tablespace

Tablespace DB33971.TL971BP1 not found on target
New LOB Tablespace DB33971.TL971BP1 will be added

Compare tablespace source(DB33971.TS33971A) and target(DB33971.TS33971A)
(A)Field Numparts changed from 3 to 4
Tablespace will be altered

Tablespace DB33971.TS33971B not found on target
New Tablespace DB33971.TS33971B will be added

Compare table source(VNDR1.TB33971A) and target(VNDR1.TB33971A)
(D)Column CLOB2 added

(A)Partition(s) added to the target table

Auxiliary table VNDR1.TB971AP1 processed

Auxiliary table VNDR1.TB971AP2 processed

Auxiliary table VNDR1.TB971AP3 processed

Auxiliary table VNDR1.TB971APN added

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 and with 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.

Table VNDR1.TB33971A is partitioned and will be dropped by dropping tablespace DB33971.TS33971A

Table will be recreated

Table data will not be converted

Not eligible for FORMAT INTERNAL processing

Table VNDR1.TB33971B not found on target

Auxiliary table VNDR1.TB971BP1 processed

ADB7150W Source contains incomplete set of explicit LOB objects therefore all LOB objects for this base table will be created implicitly.

New Table VNDR1.TB33971B will be added

Compare index source(VNDR1.IX33971A) and target(VNDR1.IX33971A)
Index VNDR1.IX33971A will be dropped by dropping the tablespace
Index will be recreated because the base table will be dropped and recreated

Index VNDR1.IX33971B not found on target
New Index VNDR1.IX33971B will be added

Index VNDR1.IX971APN not found on target
This is an index on auxiliary table.
The index is will not be created because the auxiliary table has been converted from explicit to implicit.
The respective index will be created implicitly by DB2.

Compare auxiliary index source(VNDR1.IX971AP1) and target(VNDR1.IX971AP1)
Index VNDR1.IX971AP1 will be dropped
Index will not be recreated because the auxiliary table has been converted from explicit to implicit

Compare auxiliary index source(VNDR1.IX971AP2) and target(VNDR1.IX971AP2)
Index VNDR1.IX971AP2 will be dropped
Index will not be recreated because the auxiliary table has been converted from explicit to implicit

Compare auxiliary index source(VNDR1.IX971AP3) and target(VNDR1.IX971AP3)
Index VNDR1.IX971AP3 will be dropped

Index will not be recreated because the auxiliary table has been converted from explicit to implicit

Index VNDR1.IX971BP1 not found on target

This is an index on auxiliary table.

The index is will not be created because the auxiliary table has been converted from explicit to implicit.

The respective index will be created implicitly by DB2.

COMPARISON SUMMARY REPORT

=====

| Obtyp | Source Object | Target Object | Result | Object type |
|-------|------------------|------------------|-----------------|-------------|
| D | DB33971 | DB33971 | No change | Database |
| S | DB33971.TL971APN | | Not added | Tablespace |
| T | VNDR1.TB971APN | | Not added | Table |
| X | VNDR1.IX971APN | | Not added | Index |
| S | DB33971.TL971AP1 | DB33971.TL971AP1 | No change | Tablespace |
| T | VNDR1.TB971AP1 | VNDR1.TB971AP1 | Dropped | Table |
| X | VNDR1.IX971AP1 | VNDR1.IX971AP1 | Dropped | Index |
| S | DB33971.TL971AP2 | DB33971.TL971AP2 | No change | Tablespace |
| T | VNDR1.TB971AP2 | VNDR1.TB971AP2 | Dropped | Table |
| X | VNDR1.IX971AP2 | VNDR1.IX971AP2 | Dropped | Index |
| S | DB33971.TL971AP3 | DB33971.TL971AP3 | No change | Tablespace |
| T | VNDR1.TB971AP3 | VNDR1.TB971AP3 | Dropped | Table |
| X | VNDR1.IX971AP3 | VNDR1.IX971AP3 | Dropped | Index |
| S | DB33971.TL971BP1 | | Not added | Tablespace |
| T | VNDR1.TB971BP1 | | Not added | Table |
| X | VNDR1.IX971BP1 | | Not added | Index |
| S | DB33971.TS33971A | DB33971.TS33971A | Altered | Tablespace |
| T | VNDR1.TB33971A | VNDR1.TB33971A | Dropped/created | Table |
| X | VNDR1.IX33971A | VNDR1.IX33971A | Dropped/created | Index |
| S | DB33971.TS33971B | | Added | Tablespace |
| T | VNDR1.TB33971B | | Added | Table |
| X | VNDR1.IX33971B | | Added | Index |

COMPARISON COUNTS REPORT

=====

| Object type | On source | On target | Compared | Added | Dropped | Altered | Not added |
|-------------------|-----------|-----------|----------|-------|---------|---------|-----------|
| Schemas | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| User Def Types | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Sequences | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Databases | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| Tablespaces | 7 | 4 | 4 | 1 | 0 | 1 | 2 |
| Tables | 7 | 4 | 4 | 1 | 0 | 0 | 2 |
| Indexes | 7 | 4 | 4 | 1 | 0 | 0 | 2 |
| Aliases | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Storage groups | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Synonyms | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Functions | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Stored procedures | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Triggers | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Views | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Column masks | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row permissions | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Relations | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

The count report groups all objects by the type and reports the number of objects on the source and on the target. The count report also lists the number of objects compared, added or not added (on source only), and dropped (if on target only). You can also see how many objects from the compared objects were altered.

Compare DB2 Objects sample summary conversion report

You might have few change windows to work with and limited time to run Work Statement Lists. You need a quick way to determine whether conversions will take place when a change is run, so you know what the potential problems are. The sample summary conversion report in the following figure shows the truncations and conversions that will take place when the change runs. This report is printed on a separate DD-card (CONVRPT).

***** TOP OF DATA *****

GOC2CMP – Expected Conversion Errors

Database 2 Object Comparison Tool
5697-L40 Copyright IBM Corporation 2001, 2009.
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.

GOC2CMP – Expected Conversion Errors

Source:
 Extracted from location *FROM DDL FILE* at 2009-06-23 13:52 by VNDRG

Target:
 Extracted from DSN9 at 2009-06-23 13:52 by VNDRG

Target system is DB2 Release 915

Conversion report generated in ANALYZE mode

CONVERSION REPORT SUMMARY
=====

Compare table source(VNDRG.SRC257TB) and target(VNDRG.TAR257TB)
Column name From type To type

EMPNO INTEGER SMALLINT
NAME CHAR(30) CHAR(25)

Compare table source(VNDRG.SRC257TB1) and target(VNDRG.TAR257TB1)
Column name From type To type

SALARY SMALLINT DECIMAL(5,2)

***** BOTTOM OF DATA *****

Possible conversion errors

This topic provides a summary list of possible conversion errors that might occur due to truncation during conversion of data types.

Possible conversion errors

The following list shows possible conversion errors.

Integer to Smallint

Integer to Decimal
Smallint to Decimal
Float to Smallint
Float to Integer
Float to Bigint
Float to decimal
Decimal to Smallint
Decimal to Integer
Decimal to Decimal
Decimal to Date
Decimal to Time
Decimal to Timestamp
Char to Char
Char to Binary
Char to Varchar
Char to Date
Char to Time
Char to Timestamp
Char to Smallint
Char to Integer
Char to decimal
Char to Longvar
Binary to Binary
Varbinary to Varbinary
Varchar to Char
Varchar to Varchar
Varchar to Varbinary
Varchar to Time
Varchar to Timestamp
Varchar to Smallint
Varchar to Integer
Varchar to Decimal
Varchar to Longvar
Varchar to Date
Longvar to Char
Longvar to Varchar
Longvar to Date
Longvar to Time
Longvar to Timestamp
Longvar to Longvar
Graphic to Graphic
Graphic to Vargraphic
Graphic to Longvarg
Vargraphic to Graphic
Vargraphic to Longvarg

Vargraphic to Vargraphic
LongVarg to Graphic
Longvarg to Vargraphic
Longvarg to Longvarg
Decfloat to Smallint
Decfloat to Integer
Decfloat to Float
Decfloat to Decimal
Decfloat to Bigint
Date to Char
Date to Varchar
Time to Char
Time to Varchar
Timestamp to Char
Timestamp to Varchar

Chapter 11. Translation masks and ignore fields

These reference topics are designed to provide you with quick access to information about DB2 Object Comparison Tool translation masks and ignore fields.

The DB2 Object Comparison Tool supports the use of translation masks so that you can compare objects with different names and provides an option to specify fields to ignore when objects are compared.

By using DB2 Object Comparison Tool, you can use masking and ignore files to handle intentional differences and different naming conventions between the objects that you are comparing. For example, primary and secondary quantities usually differ between test and production systems. Likewise, the same object might have an owner name of TESTxxx on the test system and an owner name of PRODxxx on the production system. You can use the masking and ignore files to compare only for actual differences.

Topics:

- “Translation masks”
- “Translation mask names” on page 192
- “Ignore fields” on page 194
- “Generic ignore field specifications” on page 196 “XMLMODIFIER ignore field specifications” on page 198

The following topics provide additional information:

- “Batch DB2 catalog extraction program report” on page 153

Translation masks

Use translation masks to allow a match to be found when the compare source and target objects use different naming conventions. If you specify both mask and ignore, however, the ignore overrides the mask.

Before DB2 Object Comparison Tool compares DB2 catalog record fields, masks are applied to owner and name fields.

Three types of masks are supported:

AUTHID masks

AUTHID masks are applied to all fields that contain DB2 authorization IDs (for example, OWNER and CREATOR).

NAME masks

NAME masks are applied to all fields naming objects.

RENAME specifications

RENAME specifications are applied to specify that an object in the source was renamed and should be related to an existing object in the target.

The syntax for RENAME objects is as follows:

```
renameobj:old-name,new-name
```

You can use the wildcard character, an asterisk (*), in the RENAME syntax.

You can rename the following objects:

RENAMEDB

Rename database. Example syntax: X*,P*

RENAMETS

Rename tablespace. Example syntax: X*.X*,P*.P*

RENAMETB

Rename table. Example syntax: *.X*,*.P*

RENAMEIX

Rename index. Example syntax: *.X*,*.P*

RENAMEGV

Rename global variable. Example syntax: *.GVT*,*.GVS*

An error message is generated:

- If the name of compared columns is specified as an input mask in the RENAME column mask.
- A column with the name of the output mask does not exist.

You can use one or more translation masks on the source object to make it match the target object. Translation masks can contain generic specifications, which are expressed by using an asterisk (*).

All version file (catalog) records for an object have a prefix with control information for the comparison process. Qualifier and name are parts of the prefix, and masks are applied to the source file prefix to align it with the target system naming conventions.

The syntax for specifying a mask is shown in the following figure:

| |
|--------------------------------|
| maskname: inputmask,outputmask |
|--------------------------------|

Figure 95. Translation mask syntax

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.

Several examples of mask specifications are shown in the following figure. 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 before the compare operation is performed.

Example 2:

NAME: HLQ*D*, NEW**

In this example, name HLQ47D9 is translated to NEW479 before it is compared with the target database. Masks are applied to the source before the source is compared to the target.

Example 3:

AUTHID: SYSIBM, COPY

In this example, all authorization IDs that have the value SYSIBM are translated to COPY.

Example 4:

AUTHID: *01*, *DB01*

In this example, an owner of PROD01 is translated to PRODDB01.

Example 5:

COLNAME: COL*, NEWCOL*

In this example, any column name in any table that starts with 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.

Example 6:

COMPRESS: TESTDB.TESTHRTS*, YES

In this example, the table spaces in the TESTDB database that start with TESTHRTS will be compressed.

Figure 96. Examples of translation masks

Note: If the specified data set name exists, it is reused. Otherwise, it is created.

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.

When you specify masks, they are processed in the order that you list them.

It is important to understand that masks are applied to the source objects before they are compared with the target objects.

The masks that translate names are processed first and then any masks that specify overwrite values are applied.

Performance tip: Using many masks that translate names 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.

You can use a REXX user exit to specify the overwrite value for the table space and index space attributes for DEFER, DEFINE, DSSIZE, PRIQTY, and SECQTY. 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. For details on REXX user exits, see the topic "Specifying a REXX user exit for the overwrite value" in the *DB2 Administration Tool for z/OS User's Guide and Reference*.

Masks for specific objects

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.

Table 13. Object-specific masks and the objects they affect

| Name | Syntax |
|-----------------------|---|
| ALNAME | ALNAME: <i>alias_schema.alias_name:current_alname,new_alname</i> |
| ALSCHEMA | ALSCHEMA: <i>alias_schema.alias_name:current_alschema,new_alschema</i> |
| COLNAME | COLNAME: <i>table_schema.table_name:current_colname,new_colname</i> |
| CONSNAME | CONSNAME: <i>table_schema.table_name:current_consname,new_consname</i> |
| DBOWNER | DBOWNER: <i>database_name:current_dboowner,new_dboowner</i> |
| DBRMNAME ₁ | DBRMNAME: <i>stp_schema.stp_name:current_dbrmname,new_dbrmname</i> |
| DBRMNAME | DBRMNAME: <i>udf_schema.udf_name:current_dbrmname,new_dbrmname</i> |
| DBRMNAME | DBRMNAME: <i>table_schema.table_name:current_dbrmname,new_dbrmname</i> |
| GRPNAME | GRPNAME: <i>database_name:current_grpname,new_grpname</i> |
| GVNAME | GVNAME: <i>gv_schema.gv_name:current_gvname,new_gvname</i> |
| GVSHEMA | GVSHEMA: <i>gv_schema.gv_name:current_gvschema,new_gvschema</i> |
| IXBPNAME ₂ | IXBPNAME: <i>index_schema.index_name:current_bpname,new_bpname</i> |
| IXBPNAME | IXBPNAME: <i>database_name:current_db_indexbpname,new_db_indexbpname</i> |
| IXNAME | IXNAME: <i>index_schema.index_name:current_ixname,new_ixname</i> |
| IXOWNER | IXOWNER: <i>index_schema.index_name:current_ixowner,new_ixowner</i> |
| IXSCHEMA | IXSCHEMA: <i>index_schema.index_name:current_ixschema,new_ixschema</i> |
| IXSGNAME | IXSGNAME: <i>index_schema.index_name:current_ixsgname,new_ixsgname</i> |
| LOCATION | LOCATION: <i>schema_name.obj_name:current_location,new_location</i> |
| MKNAME | MKNAME: <i>mask_schema.mask_name:current_maskname,new_maskname</i> |
| MKSCHEMA | MKSCHEMA: <i>mask_schema.mask_name:current_mkschema,new_mkschema</i> |
| PGMNAME ₁ | PGMNAME: <i>stp_schema.stp_name:current_pgmname,new_pgmname</i> |
| PGMNAME | PGMNAME: <i>udf_schema.udf_name:current_pgmname,new_pgmname</i> |
| PGMNAME | PGMNAME: <i>table_schema.table_name:current_pgmname,new_pgmname</i> |
| PKGNAME | PKGNAME: <i>collection_id.package_name:current_pkgname,new_pkgname</i> |
| PKGOWNER | PKGOWNER: <i>collection_id.package_name:current_packageowner,new_packageowner</i> |

Table 13. Object-specific masks and the objects they affect (continued)

| Name | Syntax |
|-----------------------|--|
| PMNAME | PMNAME: <i>pm_schema.pm_name:current_pmname,new_pmname</i> |
| PMSHEMA | PMSHEMA: <i>pm_schema.pm_name:current_pmschema,new_pmschema</i> |
| SEQNAME | SEQNAME: <i>seq_schema.seq_name:current_seqname,new_seqname</i> |
| SEQSCHEMA | SEQSCHEMA: <i>seq_schema.seq_name:current_seqschema,new_seqschema</i> |
| SGOWNER | SGOWNER: <i>stogroup_name:current_stogroupowner, new_stogroupowner</i> |
| STPNAME | STPNAME: <i>stp_schema.stp_name:current_stpname,new_stpname</i> |
| STPSCHEMA | STPSCHEMA: <i>stp_schema.stp_name:current_stpschema,new_stpschema</i> |
| SYNNAME | SYNNAME: <i>synonym_owner.synonym_name:current_synname,new_synname</i> |
| SYNOWNER | SYNOWNER: <i>synonym_owner.synonym_name:current_synowner,new_synowner</i> |
| TBNAME | TBNAME: <i>table_schema.table_name:current_tbname,new_tbname</i> |
| TBOWNER | TBOWNER: <i>table_schema.table_name:current_tbower,new_tbower</i> |
| TBSCHEMA | TBSCHEMA: <i>table_schema.table_name:current_tbschema,new_tbschema</i> |
| TGNAME | TGNAME: <i>trigger_schema.trigger_name:current_tgname,new_tgname</i> |
| TGSCHEMA | TGSCHEMA: <i>trigger_schema.trigger_name:current_tgschema,new_tgschema</i> |
| TSBPNAME ₂ | TSBPNAME: <i>database_name.tablespace_name:current_tspbname,new_tspbname</i> |
| TSBPNAME | TSBPNAME: <i>database_name:current_dbbpname,new_dbbpname</i> |
| TSNAME | TSNAME: <i>database_name.tablespace_name:current_tsname,new_tsname</i> |
| TSOWNER | TSOWNER: <i>database_name.tablespace_name:current_tsowner,new_tsowner</i> |
| TSSGNAME ₂ | TSSGNAME: <i>database_name.tablespace_name:current_tssgname,new_tssgname</i> |
| TSSGNAME | TSSGNAME: <i>database_name:current_dbsgname,new_dbsgname</i> |
| UDFNAME | UDFNAME: <i>udf_schema.udf_name:current_udfname,new_udfname</i> |
| UDFSCHEMA | UDFSCHEMA: <i>udf_schema.udf_name:current_udfschema,new_udfschema</i> |
| UDTNAME | UDTNAME: <i>udt_schema.udt_name:current_udtname,new_udtname</i> |
| UDTSCHEMA | UDTSCHEMA: <i>udt_schema.udt_name:current_udtschema,new_udtschema</i> |
| VCATNAME | VCATNAME: <i>stogroup_name:current_vcatname, new_vcatname</i> |
| VCATNAME ₁ | VCATNAME: <i>schema.obj_name:current_vcatname,new_vcatname</i> |
| VWNAME | VWNAME: <i>view_schema.view_name:current_vwname,new_vwname</i> |
| VWSCHEMA | VWSCHEMA: <i>view_schema.view_name:current_vwschema,new_vwschema</i> |
| WLMENV ₁ | WLMENV: <i>udf_schema.udf_name:current_wlmenvname,new_wlmenvname</i> |
| WLMENV | WLMENV: <i>stp_schema.stp_name:current_wlmenvname,new_wlmenvname</i> |

Related concepts:

“Special considerations for comparing DB2 objects” on page 158

You can perform most comparisons field by field, comparing the catalog records that represent the objects. However, special considerations are needed in some situations.

Related tasks:

“Specifying the compare masks data set” on page 90

Use the Specify Compare Masks panel to specify the name of the data set that contains the translation masks. You can also use this panel to specify whether you want to edit that data set now.

Related reference:

“Translation mask names” on page 192

The Translation mask names table describes the translation mask names and shows their hierarchy.

Chapter 12, “DB2 catalog records and associated masks,” on page 201
The Mask application details table shows the fields in DB2 catalog records that have masks applied before the compare process takes place or before GEN creates DDL.

Translation mask names

The Translation mask names table describes the translation mask names and shows their hierarchy.

Table 14. Translation mask names

| Name | Parent | Grandparent | Description |
|----------|---------|-------------|--|
| SINGLECH | | | Single character mask specification |
| COLNAME | | | Column name |
| | | NAME | All names listed below |
| COLLNAME | | NAME | Collection name |
| CONSNAM | | 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 |
| 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 |
| SGOWNER | OWNER | AUTHID | Storage group owner |

Table 14. Translation mask names (continued)

| Name | Parent | Grandparent | Description |
|----------------------------------|--------|-------------|---|
| PKGOWNER | OWNER | AUTHID | Package owner |
| 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 |
| 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 |
| PMSHEMA | 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 |
| 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 |

Table 14. Translation mask names (continued)

| Name | Parent | Grandparent | Description |
|-----------|---------|-------------|--|
| 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 | | | |
| TSCLOSE | | | |
| IXCLOSE | | | |
| FREEPG | | | |
| 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 |
| IXERASE | ERASE | | Indicates whether the DB2-managed data sets are to be erased for indexes |

Tip: Use caution when specifying mask field **SEGSIZE: *, n**. The mask field might cause changes to the table space type. For example, specifying the mask field 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 re-created, Object Comparison Tool will generate error message ADB7139E and stop processing to avoid generating SQLCODE -662 during execution.

Related concepts:

“Translation masks” on page 187

Use translation masks to allow a match to be found when the compare source and target objects use different naming conventions. If you specify both mask and ignore, however, the ignore overrides the mask.

Ignore fields

By using ignore fields, you can compare DB2 catalog records while ignoring some fields. Ignore fields are used in situations where you are aware of differences between source and target objects, but you do not want these changes to be recognized and cause a change.

The purpose of ignoring fields during comparison 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 included by default whether they are explicitly specified on the ISPF panels.

- Protect specified fields against updates

You can specify that certain fields are to be ignored by the comparison process. 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 ignore is specified is compared. If you must recreate an object because of other changes, values for ignored fields are taken from the target version file. All other fields have values taken from the source version file.

Some catalog fields are automatically ignored by the Compare program, such as statistics, dates, and internal identifiers. As mentioned previously, these fields are called *system ignores*. You can list system ignores by using the reporting options described in Chapter 10, “Batch compare report format,” on page 171.

Important: Use caution when specifying ignore fields. If possible, use the *generic* ignore field specifications: Bufferpool, Space, Storage, and Partitioning. These specifications 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.

The syntax for specifying ignore fields is shown in the following figure:

```
objecttype: field1, field2,...,fieldn
```

where:

- *objecttype* is the DB2 catalog table name
- *fieldx* is the DB2 catalog column to be ignored

Figure 97. Ignore field syntax

Several examples of ignore field specifications are shown in the following figure:

```
SYSDATABASE: BPOOL
SYSDATABASE: INDEXBP,STGROUP
SYSTABLESPACE: BPOOL
SYSTABLEPART: PQTY,SQTY,STORNAME,VCATNAME
SYSINDEXES: INDEXSPACE
SYSINDEXPART: PQTY,SQTY,STORNAME,VCATNAME
```

Figure 98. Examples of ignore fields

When you specify fields to ignore, specify the DB2 catalog table name and which fields within it are to be ignored. In Figure 98, the first example is for a database. It shows that for SYSDATABASE, the field BPOOL is ignored when the comparison is performed.

If the compared objects originate from two different versions of DB2, they might be different because more parameters, attributes, or clauses are supported by one of the version (typically the newer version). In this case, Object Comparison Tool ignores such fields just like ignore fields that are manually entered. No differences of this kind are reported.

Important: Some values are stored in the DB2 catalog in both internal and external formats. Internal format is only understood by DB2 (not documented) and external format is suitable for input and output. Object Comparison Tool will always ignore the internal format. To ignore the value, there must be an ignore specification for the field that contains the external format of the value.

For example, SYSINDEXPART.LIMITKEY keeps the high value of the limit key of the partition in internal format. SYSTABLEPART.LIMITKEY keeps the high value of the partition in external format. Therefore, if you need to ignore SYSINDEXPART.LIMITKEY, specify SYSTABLEPART.LIMITKEY.

Related tasks:

“Specifying compare ignore fields” on page 95

Specify compare ignore fields when you want to ignore some fields when comparing DB2 catalog records. Use ignore fields when there are differences in source and target objects, but you don't want the compare process to change them.

Generic ignore field specifications

Generic ignore field specifications provide a shortcut for ignoring all bufferpools, allocated space information, how data is stored, or how data is partitioned.

The generic ignore specifications are:

- Bufferpool
- Space
- Storage
- Partitioning
- Key targets
- Business time
- System time

- Source pending changes
- Row permissions
- Column masks
- Include columns
- PBG numparts
- Hash organization
- XML modifier

Specifying a generic ignore specification has the same effect as specifying the ignore fields manually. For example, if you specify the BUFFERPOOL generic ignore specification, all of the following DB2 catalog table fields are ignored:

- SYSDATABASE.BPOOL
- SYSDATABASE.INDEXBP
- SYSINDEXES.BPOOL
- SYSTABLESPACE.BPOOL

If you specify the SPACE generic ignore specification, all of the following DB2 catalog table fields are ignored:

- SYSINDEXPART.PQTY
- SYSINDEXPART.SQTY
- SYSINDEXPART.FREEPAGE
- SYSINDEXPART.PCTFREE
- SYSINDEXPART.SECQTYI
- SYSTABLEPART.PQTY
- SYSTABLEPART.SQTY
- SYSTABLEPART.FREEPAGE
- SYSTABLEPART.PCTFREE
- SYSTABLEPART.SECQTYI
- SYSTABLESPACE.MAXROWS

If you specify the STORAGE generic ignore specification, all of the following DB2 catalog table fields are ignored as well as all of the fields that are ignored by the SPACE generic ignore information:

- SYSDATABASE.STGROUP
- SYSINDEXPART.STORTYPE
- SYSINDEXPART.STORNAME
- SYSINDEXPART.VCATNAME
- SYSTABLEPART.STORTYPE
- SYSTABLEPART.STORNAME
- SYSTABLEPART.VCATNAME
- SYSVOLUMES.VOLID
- SYSSTOGROUP.VCATNAME

If you specify the PARTITIONING generic ignore specification, all of the following DB2 catalog table fields are ignored:

- 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

If you specify the KEY TARGETS generic ignore specification, the key targets from extended indexes will not be compared. The ignored values will be taken from target indexes. The column definition and the key-expression definition from the old index will be used in the CREATE INDEX statement. In addition, SYSINDEXES.KEYTARGET_COUNT and SYSINDEXES.IX_EXTENSION_TYPE will be ignored because they are related to the extended index definition.

If you specify the PBG_NUMPARTS generic ignore specification, the following DB2 catalog table field is ignored: SYSTABLESPACE.PARTITIONS.

If you specify the SOURCE_PENDING_CHANGES generic ignore specification, source pending changes are not propagated to the source object definition (they will be omitted)

If you specify the ROW_PERMISSIONS generic ignore specification, the row permissions will not be compared, added or dropped (they will be not processed).

If you specify the COLUMN_MASKS generic ignore specification, the column masks permissions will not be compared, added or dropped (they will be not processed).

If you specify the INCLUDE_COLUMNS generic ignore specification, the include columns permissions will not be compared, added or dropped (they will be not processed).

XMLMODIFIER ignore field specifications

XMLMODIFIER ignore field specifications provide a shortcut for ignoring all element names, schema locations, target namespaces, XSR object names, or XSR object schemas.

If XMLMODIFIER ignore specifications are selected, the differences in source and target are ignored and not compared. The XMLMODIFIER ignore specifications are:

Table 15. XMLMODIFIER ignore specifications

| Column name | Data type | Description |
|-----------------|-----------------------|---|
| XSROBJECTSCHEMA | VARCHAR(128) NOT NULL | Qualifier of the XML schema name. This is always set to 'SYSXSR'. |
| XSROBJECTNAME | VARCHAR(128) NOT NULL | Name of the XML schema. |

Table 15. XMLMODIFIER ignore specifications (continued)

| Column name | Data type | Description |
|-----------------|------------------|--|
| TARGETNAMESPACE | INTEGER | The value of the STRINGID column in SYSIBM.SYSXMLSTRINGS when the target namespace URI of the primary XML schema document is stored in SYSIBM.SYSXMLSTRINGS. |
| SCHEMALOCATION | INTEGER | The value of the STRINGID column in SYSIBM.SYSXMLSTRINGS when the schema location URI of the primary XML schema document is stored in SYSIBM.SYSXMLSTRINGS. |
| ELEMENTNAME | INTEGER NOT NULL | String id for the local name of the root element node. It would be 0 if it is not specified. The value of element name stored in SYSIBM.SYSXMLSTRINGS. |

Chapter 12. DB2 catalog records and associated masks

The Mask application details table shows the fields in DB2 catalog records that have masks applied before the compare process takes place or before GEN creates DDL.

Table 16. Mask application details. The following table shows the catalog column and the most specified mask name for each catalog records that has a mask applied before the compare process takes place or before GEN creates DDL

| DB2 Catalog record | Catalog column | Most specific mask names | Comments |
|--------------------|----------------|--------------------------|-----------------------------------|
| SYSAUXRELS | TBNAME | TBNAME | |
| | TOWNER | OWNER | |
| | | TOWNER | |
| | COLNAME | COLNAME | |
| | AUXTBNAME | TBNAME | |
| | AUXTOWNER | OWNER | |
| TOWNER | | | |
| SYSCHECKS | TOWNER | OWNER | |
| | | TOWNER | |
| | CREATOR | OWNER | |
| | TBNAME | TBNAME | |
| | CHECKCONDITION | COLNAME | Mask column names |
| SYSCOLAUTH | GRANTOR | GRANTOR | |
| | TNAME | TBNAME | |
| | CREATOR | OWNER | |
| | | TOWNER | |
| | 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 | |

Table 16. Mask application details (continued). The following table shows the catalog column and the most specified mask name for each catalog records that has a mask applied before the compare process takes place or before GEN creates DDL

| DB2 Catalog record | Catalog column | Most specific mask names | Comments |
|--------------------|----------------|--------------------------|--|
| SYSCOLUMNS | NAME | COLNAME | |
| | TBNAME | TBNAME | |
| | TBCREATOR | OWNER | If schema not SYSIBM |
| | | TBOWNER | If schema not SYSIBM |
| | TYPENAME | UDTNAME | If schema not SYSIBM |
| | TYPESHEMA | 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 | PMSHEMA | If control_type is row permission |
| | NAME | PMNAME | |
| | SCHEMA | MKSCHEMA | If control_type is column mask |
| | | MKNAME | |
| SYSDATABASE | NAME | DBNAME | |
| | CREATOR | OWNER | |
| | | DBOWNER | |
| | STGROUP | TSSGNAME | |
| | BPOOL | TSBPNAME | |
| | GROUP_MEMBER | GRPNAME | |
| | INDEXBP | IXBPNAME | |
| SYSDATATYPES | SCHEMA | SCHEMA | |
| | OWNER | OWNER | |
| | NAME | UDTNAME | |
| | INLINE_LENGTH | DTINLOB | If distinct type is based on LOB source type |
| SYSFIELDS | TBCREATOR | OWNER | |
| | | TBOWNER | |
| | TBNAME | TBNAME | |
| | NAME | COLNAME | |
| | FLDPROC | PGMNAME | |

Table 16. Mask application details (continued). The following table shows the catalog column and the most specified mask name for each catalog records that has a mask applied before the compare process takes place or before GEN creates DDL

| DB2 Catalog record | Catalog column | Most specific mask names | Comments |
|--------------------|----------------|--------------------------|----------|
| SYSFOREIGNKEYS | CREATOR | OWNER | |
| | | TBOWNER | |
| | TBNAME | TBNAME | |
| | RELNAME | NAME | |
| | COLNAME | COLNAME | |
| SYSINDEXES | NAME | IXNAME | |
| | CREATOR | OWNER | |
| | | IXOWNER | |
| | TBNAME | TBNAME | |
| | TBCREATOR | OWNER | |
| | | TBOWNER | |
| | DBNAME | DBNAME | |
| | BPOOL | IXBPNAME | |
| | CLOSERULE | CLOSE | |
| | | IXCLOSE | |
| ERASERULE | ERASE | | |
| | IXERASE | | |
| SYSINDEXPART | FREEPAGE | FREEPG | |
| | | IXFREEPG | |
| | IXNAME | IXNAME | |
| | IXCREATOR | OWNER | |
| | | IXOWNER | |
| STORNAME | IXSGNAME | | |
| VCATNAME | VCATNAME | | |
| SYSKEYCOLUSE | TBCREATOR | OWNER | |
| | | TBOWNER | |
| | TBNAME | TBNAME | |
| COLNAME | COLNAME | | |
| SYSKEYS | IXNAME | IXNAME | |
| | IXCREATOR | OWNER | |
| | | IXOWNER | |
| COLNAME | COLNAME | | |

Table 16. Mask application details (continued). The following table shows the catalog column and the most specified mask name for each catalog records that has a mask applied before the compare process takes place or before GEN creates DDL

| DB2 Catalog record | Catalog column | Most specific mask names | Comments |
|--------------------|----------------|--------------------------|---------------------------|
| SYSPACKAGE | COLLID | SCHEMA | Trigger package |
| | NAME | TGNAME | Trigger package |
| | COLLID | COLLNAME | Normal package |
| | NAME | PKGNAME | Normal package |
| | OWNER | PKGOWNER | |
| | CREATOR | PKGOWNER | |
| | QUALIFIER | OWNER | |
| | PATHSCHEMAS | SCHEMA | Applied to each schema |
| 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 |

Table 16. Mask application details (continued). The following table shows the catalog column and the most specified mask name for each catalog records that has a mask applied before the compare process takes place or before GEN creates DDL

| DB2 Catalog record | Catalog column | Most specific mask names | Comments |
|--------------------|----------------|--------------------------|--------------------------------|
| 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 | |
| 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' |
| | | 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 | |

Table 16. Mask application details (continued). The following table shows the catalog column and the most specified mask name for each catalog records that has a mask applied before the compare process takes place or before GEN creates DDL

| DB2 Catalog record | Catalog column | Most specific mask names | Comments |
|--------------------|----------------|--------------------------|------------------|
| 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 | |
| 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 | SGOWNER | |
| | VCATNAME | VCATNAME | |
| SYSSYNONYMS | NAME | TBNAME | |
| | CREATOR | OWNER | |
| | TBNAME | TBNAME | |
| | TBCREATOR | OWNER | |
| | | TBOWNER | |

Table 16. Mask application details (continued). The following table shows the catalog column and the most specified mask name for each catalog records that has a mask applied before the compare process takes place or before GEN creates DDL

| DB2 Catalog record | Catalog column | Most specific mask names | Comments |
|--------------------|----------------|--------------------------|-------------------|
| 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 | | |
| SYSTABLEPART | TSNAME | TSNAME | |
| | DBNAME | DBNAME | |
| | IXNAME | IXNAME | |
| | IXCREATOR | OWNER | |
| | | IXOWNER | |
| | STORNAME | TSSGNAME | |
| | VCATNAME | VCATNAME | |
| | COMPRESS | COMPRESS | |
| | HASHSPACE | HASHSPC | |
| | TRACKMOD | TRACKMOD | |
| | | PQTY | PRIQTY |
| | SQTY | TSPRIQTY | |
| | | SECQTY | |
| | FREEPAGE | TSSECQTY | |
| | | FREEPG | |
| PCTFREE | TSFREEPG | | |
| | PCTFREE | | |
| | TSPCTFREE | | |

Table 16. Mask application details (continued). The following table shows the catalog column and the most specified mask name for each catalog records that has a mask applied before the compare process takes place or before GEN creates DDL

| DB2 Catalog record | Catalog column | Most specific mask names | Comments |
|--------------------|----------------|--------------------------|----------|
| SYSTABLES | NAME | TBNAME | |
| | CREATOR | OWNER | |
| | | TBOWNER | |
| | DBNAME | DBNAME | |
| | TSNAME | TSNAME | |
| | EDPROC | PGMNAME | |
| | VALPROC | PGMNAME | |
| | TBCREATOR | OWNER | |
| | | TBOWNER | |
| | TBNAME | TBNAME | |
| | AUDITING | AUDIT | |
| DATACAPTURE | DCAPTURE | | |
| SYSTABLESPACE | NAME | TSNAME | |
| | CREATOR | OWNER | |
| | | TSOWNER | |
| | DBNAME | DBNAME | |
| | BPOOL | TSBPNAME | |
| | SEGSIZE | SEGSIZE | |
| | LOCKMAX | LOCKMAX | |
| | CLOSERULE | CLOSE | |
| TSCLOSE | | | |
| ERASERULE | ERASE | | |
| | TSERASE | | |

Table 16. Mask application details (continued). The following table shows the catalog column and the most specified mask name for each catalog records that has a mask applied before the compare process takes place or before GEN creates DDL

| DB2 Catalog record | Catalog column | Most specific mask names | Comments |
|--------------------|----------------|--------------------------|------------------------|
| 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 | |
| SYSVARIABLES | SCHEMA | GVSHEMA | |
| | NAME | GVNAME | |
| | OWNER | OWNER | |
| SYSVARIABLEAUTH | GRANTOR | GRANTOR | |
| | GRANTEE | GRANTEE | |
| | SCHEMA | GVSHEMA | |
| | NAME | GVNAME | |
| 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 | |

Table 16. Mask application details (continued). The following table shows the catalog column and the most specified mask name for each catalog records that has a mask applied before the compare process takes place or before GEN creates DDL

| DB2 Catalog record | Catalog column | Most specific mask names | Comments |
|---------------------------|-----------------------|---------------------------------|-----------------|
| SYSVOLUMES | SGNAME | TSSGNAME | |
| | SGCREATOR | OWNER | |
| XSROBJECTS | XSROBJECTNAME | XMLSCHID | |

Related concepts:

“Translation masks” on page 187

Use translation masks to allow a match to be found when the compare source and target objects use different naming conventions. If you specify both mask and ignore, however, the ignore overrides the mask.

Related tasks:

“Specifying the compare masks data set” on page 90

Use the Specify Compare Masks panel to specify the name of the data set that contains the translation masks. You can also use this panel to specify whether you want to edit that data set now.

Chapter 13. Specifying alternate input to the generate apply job program

You can specify data sets for DB2 Object Comparison Tool to use as alternate inputs to the program that generates the apply job. In addition, you can create a template that specifies the batch parameter variables you want your data set to contain.

About this task

There are two members in the primary input data set that store primary input variables: GOCSVARS and GOCSVAR2. You can instruct the generate apply job program to use the alternate data sets by adding a DD statement to the JCL. The alternate data sets can contain variables with customized values.

Procedure

1. Create a primary input data set.
 - a. Select option 5 on the DB2 Object Comparison Tool menu to generate the data set that is referenced by the GOCSVARS DD statement.
2. Create new data sets based on the primary input data set.
 - a. Enter an I in the option field on the DB2 Object Comparison Tool menu. The I option is hidden and is not listed as an option on the menu. The List ISPF Table Extension Variables panel is displayed, as shown in the following figure:

```
DB2 Admin ----- List ISPF Table Extension Variables ----- 09:02

Enter/verify the following:
Data Set Name  ===>
Member Name    ===>
```

Figure 99. List ISPF Table Extension Variables panel (ADB2IIT)

- b. Enter the primary data set name and the member name (GOCSVARS for DB2 Object Comparison Tool) that you want to list.
 - c. Press Enter. The list of variables and values for the specified data set and member is displayed.
 - d. Copy the content of the member to the newly created alternate data set.

Requirement: The alternate input data set must exist prior to this step.

- e. Edit the variables listed in the newly copied alternate data set with the alternate values that you want to use as input to the generate apply job program.

Requirement: The alternate input data set must have a fixed record length of 80 characters with no sequence numbers. Each logical record begins in column one. Logical records continue on subsequent lines if they exceed the line length. Each logical record must end with a semicolon (;). All variables that are listed must exist in the alternate input data set.

- f. Save the modified variable list. While in the edit session, use the REPLACE command to save your changes.

- g. Repeat the steps above, this time entering GOCSVAR2 for member name.
3. Add a DD statement, ALTSHV, that refers to the two input members in the alternate input data set. You must specify the member name explicitly. Here is an example of the amended JCL:

```
//GOCSVARS DD DISP=SHR,DSN=HLQ.PRIMARY.SHV
//ALTSHV DD DISP=SHR,DSN=HLQ.ALTERNATE.ALTPDS(GOCSVARS)
// DD DISP=SHR,DSN=HLQ.ALTERNATE.ALTPDS(GOCSVAR2)
//CHANGES DD DISP=SHR,
// DSN=HLQ.THISCHG.CHG
```

4. Delete or rename members GOCSVARS and GOCSVAR2 from the primary input data set that is referenced by the GOCSVARS DD statement.

Alternate values for the generate apply program

The Alternate shared variable input data table provides a list of alternate shared data variable names and their meanings. Panel names that are the source of primary input values are identified in parentheses, where applicable.

Requirement: The variable names for the UNLOAD and LOAD utilities marked by an asterisk (*) in the table are required and cannot be changed. These variables must display in the alternate input data set as shown in the following example:

```
USU01=;
USU02=;
USU03=;
```

Table 17. Alternate shared variable input data

| Variable | Definition | Valid Input |
|----------|--|--|
| AAPFLIBR | DB2 Admin APF authorized library. | A dataset name. For example: HLQ.SADBLINK. |
| ADB081CM | DB2 release - V8 CM | Y or N. Specify Y if DB2 is at this release level or higher. |
| ADB081NF | DB2 release - V8 NFM | Y or N. Specify Y if DB2 is at this release level or higher. |
| ADB091CM | DB2 release - V9 CM | Y or N. Specify Y if DB2 is at this release level or higher. |
| ADB091NF | DB2 release - V9 NFM | Y or N. Specify Y if DB2 is at this release level or higher. |
| ADB101CM | DB2 release - V10 CM | Y or N. Specify Y if DB2 is at this release level or higher. |
| ADB101NF | DB2 release - V10 NFM | Y or N. Specify Y if DB2 is at this release level or higher. |
| ADB111CM | DB2 release - V11 CM | Y or N. Specify Y if DB2 is at this release level or higher. |
| ADB111NF | DB2 release - V11 NFM | Y or N. Specify Y if DB2 is at this release level or higher. |
| ADB25TUA | Template usage (ADB25TU) | Y or N. |
| ADB27ACF | Percent increase for converted data sets | An integer. |
| ADB2CPS | Catalog copy plan suffix. | A two-character alphanumeric value. |
| ADB2USM1 | Modify indicator (ADB utilities) | Y or N. |
| ADBADATA | Flag to indicate building work statement list for recovery by using the original data. | O or E. Use O to specify Original or E to specify Existing. |
| ADBANID | Analyzed change identifier | An Integer. Change ID from ADBC Prerequisite table. |
| ADBASUSB | Use trusted context in batch | YES or NO. |
| ADBASUSR | Use trusted context | AS USER value. |
| ADBBINDE | Bind error (ADBTEP2) | MAXE, SAVE, or IGNORE. |
| ADBBLKS | Blocksize (ADB2UPA) | An integer. |
| ADBELIB | Admin exec library concatenation. | A list of data set names. For example: 'HLQ.SGOCEXEC' 'HLQ.SADBEXEC'. |
| ADBJ1 | Job card line 1 (ADB2UPA). | A job card of up to 72 characters. Any valid job card syntax line. |
| ADBJ2 | Job card line 2 (ADB2UPA) | A job card of up to 72 characters. Any valid job card syntax line pt2. |
| ADBJ3 | Job card line 3 (ADB2UPA) | A job card of up to 72 characters. Any valid job card syntax line pt3. |

Table 17. Alternate shared variable input data (continued)

| | | |
|----------|---|---|
| ADBJ4 | Job card line 4 (ADB2UPA) | A job card of up to 72 characters. Any valid job card syntax line pt4. |
| ADBJ5 | Job card line 5 (ADB2UPA) | A job card of up to 72 characters. Any valid job card syntax line pt5. |
| ADBJCGN | Generate Job class (ADB2UPA) | Y. Use Y to specify ADBJCLS (or DB2AJCLS if not set) as the job class. |
| ADBJCLS | Job class | A-Z, 0-9. |
| ADBJPM1 | Job parm line 1 (ADB2UPA) | Any valid /*JOBPARM card syntax. For example: SYSAFF=SY4A. |
| ADBJPM2 | Job parm line 2 (ADB2UPA) | Any valid /*JOBPARM card syntax. For example: SYSAFF=SY4A. |
| ADBJPM3 | Job parm line 3 (ADB2UPA) | Any valid /*JOBPARM card syntax. For example: SYSAFF=SY4A. |
| ADBJPM4 | Job parm line 4 (ADB2UPA) | Any valid /*JOBPARM card syntax. For example: SYSAFF=SY4A. |
| ADBJTEP2 | ADBTEP2 restart parm (ADB2UPA). | Y, N, or F. Any value other than N is interpreted as yes. (FORCE), or U (USER). |
| ADBLLIB | The Admin steplib library concatenation. | The Admin Tool load library allocation. For example: 'HLQ.SGOCLLIB' 'HLQ.SADBLLIB'; |
| ADBMXDSD | Maximum allocation to DASD (ADB2UPA) | A numeric value in kilobytes. |
| ADBMXPRI | Maximum primary allocation (ADB2UPA) | A numeric value up to 3145680. |
| ADBMXPRM | Maximum primary quantity, in kilobytes, for DASD allocation (ADB2UPA) | A numeric value up to 3145680. |
| ADBNL | New line character variable | A hex value of '0D15'x. Use the hex edit capability of the ISPF editor and vertically specify 0D15 as shown here: 000386 ADBNL= ; CCCCD70154444444 14253ED5E0000000 |
| ADBNLC | New line character variable | A hex value of '0D15'x. Use the hex edit capability of the ISPF editor and vertically specify 0D15 as shown here: 000386 ADBNLC= ; CCCCDC70154444444 142533ED5E0000000 |
| ADBPRIM | Primary space allocation (ADB2UPA) | A numeric value specified in &ADBSPEC units. |
| ADBRPM1 | (ADB2UPA) | Any valid /*JOBPARM card syntax. For example: SYSAFF=SY4A. |
| ADBRPM2 | (ADB2UPA) | Any valid /*JOBPARM card syntax. For example: SYSAFF=SY4A. |
| ADBRPM3 | (ADB2UPA) | Any valid /*JOBPARM card syntax. For example: SYSAFF=SY4A. |
| ADBRPM4 | (ADB2UPA) | Any valid /*JOBPARM card syntax. For example: SYSAFF=SY4A. |
| ADBSECU | Secondary space allocation (ADB2UPA) | A numeric value specified in &ADBSPEC units. |
| ADBSPEC | Space allocation unit (ADB2UPA) | BLK, TRK, CYL or 4096-32760. |
| ADBTAPU | Tape unit (ADB2UPA) | Unit to use if allocation memory exceeds ADBMXDSD value. Esoteric name, such as 'TAPE'. |
| ADBTEPAC | Auto check (ADBTEP2) | YES, Y, NO, or N. |
| ADBTEPAI | Auto rebuild (ADBTEP2) | YES, Y, NO, or N. |
| ADBTEPAR | Auto reorg (ADBTEP2) | YES, Y, NO, or N. |
| ADBTEPCD | Check at Drop (ADBTEP2) | YES, Y, NO, or N. |
| ADBTEPIB | Advisory auto rebuild (ADBTEP2) | YES, Y, NO, or N. |
| ADBTEPIR | Advisory auto reorg (ADBTEP2) | YES, Y, NO, or N. |
| ADBTEPSP | SPANNED | YES or NO. Use YES to specify SPANNED YES for utility statements or NO to specify SPANNED NO. |
| ADBTTEST | Use test plan | YES or any other value. |
| ADBTTLB | Template library name (ADB25TU) | The ISPF table name defined by ADBGAJOB if online processing or "Y" if batch processing. |
| ADBTSTPN | Test plan name | A name. |
| ADBUNIT | Unit (ADB2UPA) | An esoteric name, such as 'SYSALLDA'. |

Table 17. Alternate shared variable input data (continued)

| | | |
|----------|---|--|
| ADBWLDSN | Work list data set name (GOC5WL) | A data set name. |
| AHPULLIB | HPU load library | A data set name. |
| ALNALTR | DDL for the altered objects (ADB25TU3) | A template name. Associated with ALALTR keyword on ADB25TU3 panel. |
| ALNCMD | DB2 commands (ADB25TU3) | A template name. Associated with ALCMD keyword on ADB25TU3 panel. |
| ALNCNC | Load control cards for the altered objects (ADB25TU3) | A template name. Associated with ALCNC keyword on ADB25TU3 panel. |
| ALNCNT | Load control cards for the original objects (ADB25TU3) | A template name. Associated with ALCNT keyword on ADB25TU3 panel. |
| ALNCREA | DDL for the created objects (ADB25TU3) | A template name. Associated with ALCREA keyword on ADB25TU3 panel. |
| ALNDROP | DDL for the dropped objects (ADB25TU3) | A template name. Associated with ALDROP keyword on ADB25TU3 panel. |
| ALNMTC | Name of non-utility data set for multi-target change information | A template name. Associated with ALMTC keyword on ADB25TU3 panel. |
| ALNRBND | DB2 commands for the rebind of plans and packages (ADB25TU3) | A template name. Associated with ALRBND keyword on ADB25TU3 panel. |
| ALNREFR | DDL for the refresh of materialized query tables (MQT) (ADB25TU3) | A template name. Associated with ALREFR keyword on ADB25TU3 panel. |
| ALNULD | Unloaded data from the original objects (ADB25TU3) | A template name. Associated with ALULD keyword on ADB25TU3 panel. |
| ALNULDC | Converted unload data (ADB25TU3) | A template name. Associated with ALULDC keyword on ADB25TU3 panel. |
| ALTDSN | Alter control card data set name (ADB25TU) | A data set name. |
| ALUALTR | Use indicator for DDL for the altered objects (ADB25TU3) | / or blank. Specify / to use, or blank to not use. |
| ALUCMD | Use indicator for DB2 commands (ADB25TU3) | / or blank. Specify / to use, or blank to not use. |
| ALUCNC | Use indicator for load control cards for the altered objects (ADB25TU3) | / or blank. Specify / to use, or blank to not use. |
| ALUCNT | Use indicator for Load control cards for the original objects (ADB25TU3) | / or blank. Specify / to use, or blank to not use. |
| ALUCREA | Use indicator for DDL for the created objects (ADB25TU3) | / or blank. Specify / to use, or blank to not use. |
| ALUDROP | Use indicator for DDL for the dropped objects (ADB25TU3) | / or blank. Specify / to use, or blank to not use. |
| ALUMTC | Use indicator for non-utility multi-target template (ADB25TU) for multi-target change | / or blank. Specify / to use, or blank to not use. |
| ALURBND | Use indicator for DB2 commands for the rebind of plans and packages (ADB25TU3) | / or blank. Specify / to use, or blank to not use. |
| ALUREFR | Use indicator for DDL for the refresh of materialized query tables (MQT) (ADB25TU3) | / or blank. Specify / to use, or blank to not use. |
| ALUULD | Use indicator for Unloaded data from the original objects (ADB25TU3) | / or blank. Specify / to use, or blank to not use. |
| ALUULDC | Use indicator for converted unload data (ADB25TU3) | / or blank. Specify / to use, or blank to not use. |
| ASYRECD | Use activate HPU Parallel Unload/Load in the batch apply job (ADB2UCUS) | A template name. Associated with ALULD keyword on ADB2UCUS panel. |
| ASYREDCD | Use activate HPU Parallel Unload/Load in the batch apply job (ADB2UCUS) | A template name. Associated with ALULD keyword on ADB2UCUS panel. |
| ASYSLIA | ISPF linklist library 2 | A data set name. |
| ASYSLIB | ISPF linklist library 1 | A data set name. |
| ASYMLIB | ISPF message library | A data set name. |
| ASYTLIB | ISPF table library | A data set name. |
| CLOBCOLN | LOBCOLDDN (ADB25TU4) | A template name. |
| CLOBCOLU | Use indicator for LOBCOLDDN (ADB25TU4) | / or blank. Specify / to use, or blank to not use. |
| CREATDSN | Create control card data set name (ADB25TU) | A data set name. |
| CTNCOPY1 | COPYDDN 1 (ADB25TU4) | A template name. Used as the first parameter to the COPYDDN keyword. For example: COPYDDN(mytemp1). |
| CTNCOPY2 | COPYDDN 2 (ADB25TU4) | A template name. Used as the second parameter to the COPYDDN keyword. For example: COPYDDN(mytemp1,mytemp2). |

Table 17. Alternate shared variable input data (continued)

| | | |
|----------|--|--|
| CTNDISC1 | DISCARDN (ADB25TU4) | A template name. Used as the parameter to the DISCARDN keyword. For example: DISCARDN(<i>mytemp3</i>). |
| CTNERR | ERRDDN (ADB25TU4) | A template name. Used as the parameter to the ERRDDN keyword. For example: ERRDDN(<i>mytemp4</i>). |
| CTNFCOPY | FCCOPYDDN (ADB25TU4) | A template name. Used as the parameter to the FCCOPYDDN keyword. For example: FCCOPYDDN(<i>mytemp5</i>). |
| CTNFILTR | FILTERDDN (ADB25TU4) | A template name. Used as the parameter to the FILTERDDN keyword. For example: FILTERDDN(<i>mytemp6</i>). |
| CTNMAPDD | MAPDDN (ADB25TU4) | A template name. Used as the parameter to the MAPDDN keyword. For example: MAPDDN (<i>mytemp7</i>). |
| CTNPUNCH | PUNCHDDN (ADB25TU4) | A template name. Used as the parameter to the PUNCHDDN keyword. For example: PUNCHDDN(<i>mytemp8</i>). |
| CTNRECV1 | RECOVERYDDN 1 (ADB25TU4) | A template name. Used as the first parameter to the RECOVERYDDN keyword. For example: RECOVERYDDN(<i>mytemp9</i>). |
| CTNRECV2 | RECOVERYDDN 2 (ADB25TU4) | A template name. Used as the second parameter to the RECOVERYDDN keyword. For example: RECOVERYDDN(<i>mytemp9, mytempA</i>). |
| CTNUNLDD | UNLDDN (ADB25TU4) | A template name. Used as the parameter to the UNLDDN keyword. For example: UNLDDN(<i>mytempB</i>). |
| CTNWORK1 | WORKDDN 1 (ADB25TU4) | A template name. Used as the first parameter to the WORKDDN keyword. For example: WORKDDN(<i>mytempC</i>). |
| CTNWORK2 | WORKDDN 2 (ADB25TU4) | A template name. Used as the second parameter to the WORKDDN keyword. For example: WORKDDN(<i>mytempC, mytempD</i>). |
| CTUCOPY1 | Use indicator for COPYDDN 1 (ADB25TU4) | / or blank. Specify / to use, or blank to not use. |
| CTUCOPY2 | Use indicator for COPYDDN 2 (ADB25TU4) | / or blank. Specify / to use, or blank to not use. |
| CTUDISC1 | Use indicator for DISCARDN (ADB25TU4) | / or blank. Specify / to use, or blank to not use. |
| CTUERR | Use indicator for ERRDDN (ADB25TU4) | / or blank. Specify / to use, or blank to not use. |
| CTUFCOPY | Use indicator for FCCOPYDDN (ADB25TU4) | / or blank. Specify / to use, or blank to not use. |
| CTUFILTR | Use indicator for FILTERDDN (ADB25TU4) | / or blank. Specify / to use, or blank to not use. |
| CTUMAPDD | Use indicator for MAPDDN (ADB25TU4) | / or blank. Specify / to use, or blank to not use. |
| CTUPUNCH | Use indicator for PUNCHDDN (ADB25TU4) | / or blank. Specify / to use, or blank to not use. |
| CTURECV1 | Use indicator for RECOVERYDDN 1 (ADB25TU4) | / or blank. Specify / to use, or blank to not use. |
| CTURECV2 | Use indicator for RECOVERYDDN 2 (ADB25TU4) | / or blank. Specify / to use, or blank to not use. |
| CTUUNLDD | Use indicator for UNLDDN (ADB25TU4) | / or blank. Specify / to use, or blank to not use. |
| CTUWORK1 | Use indicator for WORKDDN 1 (ADB25TU4) | / or blank. Specify / to use, or blank to not use. |
| CTUWORK2 | Use indicator for WORKDDN 2 (ADB25TU4) | / or blank. Specify / to use, or blank to not use. |
| CXMLCOLN | XMLCOLDDN (ADB25TU4) | A template name. |
| CXMLCOLU | Use indicator for XMLCOLDDN (ADB25TU4) | / or blank. Specify / to use, or blank to not use. |
| DB2AASW | Authorization switch | Y or N. |
| DB2AJCLS | Job class for DB2 utility jobs | Alphanumeric. |
| DB2ALOAD | DB2 system library concatenation | Specify a list of data sets. For example: DB2ALOAD='DB2A.SDSNEXIT' 'DB2.SDSNLOAD' ; |
| DB2APREL | DB2 release | Four characters, such as 0915 or 1015. |
| DB2ARLIB | DB2 run library | A data set name. |
| DB2ASERV | DB2 current server | SSID. |
| DB2AULIB | DB2 SDSNLOAD | A data set name. |
| DB2AUTH | DB2 authid | A User ID. |
| DB2SYS | DB2 system name | SSID. |
| DROPSDN | Drop control card data set name (ADB25TU) | A data set name. |
| GOCA1JOB | Generate one job (GOC5) | Y, N, or P (one per process). |
| GOCAJDSN | Data set for apply jobs (GOC5A) | A data set name. |

Table 17. Alternate shared variable input data (continued)

| | | |
|----------------------------|--|--|
| GOCJOB | Member prefix (GOC5) | A name. |
| GOCAPCON | Content of apply job (GOC5) | A or D. Use A to specify All or D to specify DDL only. |
| GOCDELWL | Delete WSL member before writing | Y or any character. Use Y to specify Delete or anything else to specify do not delete. |
| GOCGACHK | Run CHECK DATA (GOC5) | Y or N. |
| GOCGAIMC | Run IMAGE COPY (GOC5) | R, A, B, or N. Use R to specify Reload, A to specify Alter, B to specify Both, or N to specify None. |
| GOCGARUN | Run RUNSTATS (GOC5) | R – Reload A – Alter B – Both N – None. |
| GOCGAWL | As work statement list (GOC5) | Y or N. |
| GOCGREB | Run REBIND (GOC5) | Y or N. |
| GOCGREOR | Run REORG (GOC5) | M – Mandatory A – All relevant N – None. |
| GOCJCL | PDS for batch jobs (GOC5) | A data set name. |
| GOCMIDQL | Middle level qualifier for data sets that are created | A name. |
| GOCONL | Generate online (GOC5) | Y or N. |
| GOCPRE | Prefix for data sets (GOC5) | A data set prefix |
| GOCUNLT | Unload method (GOC5) | U, P, or H. Use U to specify Unload, P to specify parallel unload, or H to specify HPU. |
| GOCUTOP | Use utility options (GOC5) | Y or N. |
| GOCWLN | Work list name (GOC5) | A name. |
| IFFDSN | Internal version file data set name (ADB25TU) | A data set name. |
| LOBCOLN | Name of the LOB data set | A template name. This value is used as the parameter to the LOBDDN keyword. For example: LOBDDN(<i>mytempF</i>). |
| LOBCOLU | Use indicator for LOB column template. | / or blank. Specify / to use, or blank to not use &LOBCOLN. |
| MAPDBNAM | MAPPINGDATABASE, a utility option for REORG tablespace | A database name. |
| MAPOWNER | Mapping table owner (ADB2USOO) | An owner or schema. |
| MAPTBNAM | Mapping table name (ADB2USOO) | A name. |
| NSTUPROC | Number of steps in DSNUPROC | An integer (1 – 20). |
| REBDSN | Rebind control card data set name (ADB25TU) | A data set name. |
| RECOVER | Recover control card data set name (ADB25TU) | A data set name. |
| REFDSN | Refresh control card data set name (ADB25TU) | A data set name. |
| RUNLIB | AHPULLIB | A data set name. |
| XMLCOLN | Name of XML column (ADB25TU) | A template name. This value is used as the parameter to the XMLDDN keyword. For example: XMLDDN(<i>mytempE</i>). |
| XMLCOLU | Use indicator for XML column template (ADB25TU) | / or blank. Specify / to use, or blank to not use &XMLCOLN |
| IMAGE COPY Utility Options | | |
| USC01 | FULL | Y or N. Use Y to specify FULL YES or N to specify FULL NO. |
| USC02 | CHANGE LIMIT | Y or A. Use A to specify CHANGELIMIT (ANY) or Y to specify CHANGELIMIT (&USC03). |
| USC03 | PERCENT VALUE1 | 0.0 to 100.0. This value is used as the first parameter to the CHANGELIMIT keyword. |
| USC04 | PERCENT VALUE2 | 0.0 to 100.0. This value is used as the second parameter to the CHANGELIMIT keyword. |
| USC05 | REPORT ONLY | Y or N. Use Y to specify REPORTONLY or N to specify no keyword. |
| USC06 | PARALLEL | YES or an integer value between 0 and 32767. |
| USC07 | CHECKPAGE | Y or N. Use Y to specify CHECKPAGE or N to specify no keyword. |
| USC08 | CONCURRENT | Y or N. Use Y to specify CONCURRENT or N to specify no keyword. |

Table 17. Alternate shared variable input data (continued)

| | | |
|----------------------------|-----------------------------|---|
| USC09 | SHRLEVEL | R or C. Use R to specify SHRLEVEL REFERENCE or C to specify SHRLEVEL CHANGE. |
| USC10 | CLONE | Y or N. Use Y to specify CLONE or N to specify no keyword. |
| USC11 | SCOPE | A or P. Use A to specify SCOPE ALL or P to specify SCOPE PENDING. |
| USC113 | TAPEUNITS | A numeric value. This value is used as a parameter to TAPEUNITS. For example: TAPEUNITS 3. |
| USC12 | FLASHCOPY | Y, N, or C. Use Y to specify FLASHCOPY YES, N to specify FLASHCOPY NO, or C to specify FLASHCOPY CONSISTENT. |
| USC123 | SYSTEMPAGES | YES or NO. |
| CHECK DATA Utility Options | | |
| USK01 | SCOPE | P, X, A, R, or M. Use P to specify SCOPE PENDING, X to specify SCOPE AUXONLY, A to specify SCOPE ALL, R to specify SCOPE REFONLY, or M to specify SCOPYE XMLSCHEMAONLY. |
| USK02 | AUXERROR | R or I. Use R to specify AUXERROR REPORT or I to specify AUXERROR INVALIDATE. |
| USK03 | EXCEPTIONS | 0-32767. This number is used as a parameter to EXCEPTIONS. For example: EXCEPTIONS 257. |
| USK04 | SORTDEVT | A device type. This value is used as a parameter to SORTDEVT. For example: SORTDEVT <i>devtype</i> . |
| USK05 | SORTNUM | 1-255. This value is used as a parameter to SORTNUM. For example: SORTNUM 93. |
| USK06 | SHRLEVEL | R or C. Use R to specify SHRLEVEL REFERENCE or C to specify SHRLEVEL CHANGE. |
| USK07 | CLONE | Y or N. Use Y to specify the CLONE keyword or N to specify no keyword. |
| USK08 | LOBERROR | R or I. Use R to specify LOBERROR REPORT or I to specify LOBERROR INVALIDATE. |
| USK09 | XMLERROR | R or I. Use R to specify XMLERROR REPORT or I to specify XMLERROR INVALIDATE. |
| USK10 | DELETE | YES or NO. Use YES to specify DELETE YES or NO to specify no keywords. |
| USK11 | LOG | YES or NO. Use YES to specify LOG YES or NO to specify LOG NO. |
| USK12 | DRAIN WAITV | 1-1800. This number is used as a parameter to DRAIN_WAIT. For example: DRAIN_WAIT 97. |
| USK13 | RETRYV | 0-255. This number is used as a parameter to RETRY. For example: RETRY 98. |
| USK14 | RETRY DELAYV | 1-1800. This number is used as a parameter to RETRY_DELAY. For example: RETRY_DELAY 103. |
| USK15 | INCLUDE XML TABLESPACES | Name or ALL. This value is used as a parameter to TABLESPACES. For example: INCLUDE XML TABLESPACES mydb.myts. Substitute your database and tablespace for mydb.myts. For long names, also provide the table &FRTAB. Long object names are not supported. |
| USK16 | INCLUDE XML COLUMNS | Use the following syntax: TABLE myschema.mytable XMLCOLUMN mycolumn. This value is used as a parameter to TABLESPACES. For example: INCLUDE XML TABLESPACES mydb.myts TABLE myschema.mytable XMLCOLUMN mycolumn. For long names, also provide table &TTNAME. Long object names are not supported. |
| USK17 | INCLUDE XMLSCHEMA | YES or NO. Use YES to specify XMLSCHEMA keyword or NO to specify no keyword. |
| USKN1 | FOR EXCEPTION IN table name | An object name. Use this value to specify FOR EXCEPTION IN name. Should also provide table &INTABL. |
| USKN2 | USE table name | An object name. Use this value to specify USE name. Also provide the table &USTABL. |

Table 17. Alternate shared variable input data (continued)

| | | |
|------------------------|-------------------------------|---|
| USKS1 | FOR EXCEPTION IN table schema | A schema name. This value is used in conjunction with &uskn1. |
| USKS2 | USE table schema | A schema name. This value is used in conjunction with &uskn2. |
| MODIFY Utility Options | | |
| USM01 | AGE | 0-32767. This value is used as a parameter to AGE. For example: DELETE AGE(27). |
| USM02 | DATE | yyyymmdd. This date value is used as a parameter to DATE. For example: DELETE DATE(20130704). |
| USM033 | CLONE | YES or NO. Use YES to specify CLONED YES and CLONE keywords or NO to specify no keyword. |
| USM04 | LASTV | 0-32767. Use this value as a parameter to LAST. For example: RETAIN LAST(41). |
| USM05 | LOGLIMITV | YES or NO. Use YES to specify RETAIN LOGLIMIT or NO to specify no keyword. |
| USM06 | GDGLIMITV LASTV | 0-32767. This value is used as a parameter to LAST. For example: RETAIN GDGLIMIT LAST(12). |
| USM061 | GDGLIMITV | YES or NO. Use YES to specify RETAIN GDGLIMIT or NO to specify no keyword. |
| USM07 | GDGLIMITV LOGLIMITV | YES or NO. Use YES to specify RETAIN GDGLIMIT LOGLIMIT or NO to specify no keyword. |
| REORG Utility Options: | | |
| USO01 | REUSE | Y or N. Use Y to specify REUSE or N to specify no keyword. |
| USO02 | LOG | Y or N. Use Y to specify LOG YES or N to specify LOG NO. |
| USO03 | SORTDATA | Y or N. Use Y to specify SORTDATA or N to specify no keyword. |
| USO04 | NOSYSREC | Y or N. Use Y to specify NOSYSREC or N to specify no keyword. |
| USO05 | SORTKEYS | Y or N. Use Y to specify SORTKEYS or N to specify no keyword. |
| USO06 | SHRLEVEL | C, R, or N. Use C to specify SHRLEVEL CHANGE, R to specify SHRLEVEL REFERENCE, or N to specify SHRLEVEL NONE. |
| USO07 | FASTSWITCH | Y or N. Use Y to specify FASTSWITCH YES or N to specify FASTSWITCH NO. |
| USO08 | OFFPOSLIMIT | 0-65535. This value is used as a parameter to OFFPOSLIMIT. For example: OFFPOSLIMIT 1021. |
| USO09 | INDREFLIMIT | 0-65535. This value is used as a parameter to INDREFLIMIT. For example: INDREFLIMIT 201. |
| USO10 | KEEPDICTIONARY | Y or N. Use Y to specify KEEPDICTIONARY or N to specify no keyword. |
| USO11 | STATISTICS | Y or N. Use Y to specify STATISTICS TABLE (ALL) or N to specify no keyword. |
| USO12 | REPORT | Y or N. Use Y to specify REPORT YES or N to specify REPORT NO. |
| USO13 | UPDATE | A, P, S, or N. Use A to specify UPDATE ALL, P to specify UPDATE ACCESSPATH, S to specify UPDATE SPACE, or N to specify UPDATE NONE. |
| USO14 | HISTORY | A,P,S, or N. Use A to specify HISTORY ALL, P to specify HISTORY ACCESSPATH, S to specify HISTORY SPACE, or N to specify HISTORY NONE. |
| USO15 | FORCEROLLUP | Y or N. Use Y to specify FORCEROLLUP YES or N to specify FORCEFOLLUP NO. |
| USO16 | PREFORMAT | Y or N. Use Y to specify PREFORMAT or N to specify no keyword. |
| USO17 | SORTDEVT | A device type. This value is used as a parameter to SORTDEVT. For example: SORTDEVT <i>devt</i> . |
| USO18 | SORTNUM | 1 - 255. This number is used as a parameter to SORTNUM. For example: SORTNUM 3. |

Table 17. Alternate shared variable input data (continued)

| | | |
|------------|----------------------------|---|
| USO19 | DEADLINE | N, timestamp, or labeled duration expression. Use N to specify DEADLINE NONE together with a timestamp that is used as a parameter to DEADLINE. For example: DEADLINE 13:15:01. An example of a labeled duration expression is CURRENT_DATE +3 DAYS. |
| USO20 | DRAIN_WAIT | 1-1800. This value is used as a parameter to DRAIN_WAIT. |
| USO21 | RETRY | 0-255. This value is used as a parameter to RETRY. For example: RETRY 8. |
| USO22 | RETRY DELAY | 1-1800. This value is used as a parameter to RETRY_DELAY. For example RETRY_DELAY 17. |
| USO24 | MAXRO | D or numeric value. Use D to specify MAXRO DEFER or numeric value to specify MAXRO &uso24. |
| USO25 | DRAIN | W or A. Use W to specify DRAIN WRITERS or A to specify DRAIN ALL. |
| USO26 | LONGLOG | C, T, or D. Use C to specify LONGLOG CONTINUE, T to specify LONGLOG TERM, or D to specify LONGLOG DRAIN. |
| USO27 | DELAY | A numeric value. This value is used as a parameter to DELAY. For example DELAY 17. |
| USO28 | TIMEOUT | A or T. Use A to specify TIMEOUT ABEND or T to specify TIMEOUT TERM. |
| USO29 | CLONE | YES or NO. Use YES to specify CLONE or NO to specify no keyword. |
| USO30 | SCOPE | A or P. Use P to specify SCOPE PENDING or A to specify no keyword. |
| USO31 | REBALANCE | Y or N. Use Y to specify REBALANCE or N to specify no keyword. |
| USO32 | REPORTONLY | Y or N. Use Y to specify REPORTONLY or N to specify no keyword. |
| USO33 | UNLOAD | C, P, O, or E. Use C to specify UNLOAD CONTINUE, P to specify UNLOAD PAUSE, O to specify UNLOAD ONLY, or E to specify UNLOAD EXTERNAL. |
| USO34 | NOPAD | Y or N. Use Y to specify NOPAD or N to specify no keyword. |
| USO35 | FROM TABLE | An object name. Also provide table &FRNAME. |
| USO36 | AUX | YES or NO. Use YES to specify AUX YES or NO to specify AUX NO. |
| USO37 | A list of partitions. | Identifies the set of partitions that are to be reorganized. For example: 1, 3, 5:8 |
| USO38 | FLASHCOPY | Y, C, or N. Use Y to specify FLASHCOPY YES, C to specify FLASHCOPY CONSISTENT, or N to specify FLASHCOPY NO. |
| USO363 | This variable is not used. | |
| I USO40 | LOGRANGES | <ul style="list-style-type: none"> • Y - Yes, REORG uses SYSLGRNX information for the LOG phase whenever possible. This option is the default behavior. • N - NO, REORG does not use SYSLGRNX information for the LOG phase. |
| I USO41 | DRAIN_ALLPARTS | <ul style="list-style-type: none"> • Y - YES, REORG obtains the table space level drain on the entire partitioned table space first, before draining the target data partitions and the indexes. • N - NO, REORG drains the target data partitions serially followed by the non-partitioned secondary indexes. This option is the default behavior. |
| I USO42 | SWITCHTIME | <ul style="list-style-type: none"> • N - NONE, does not specify a time for the final log iteration of the LOG phase. This option is the default behavior. Specifies the time that the final log iteration of the LOG phase is to begin. This time must not have already occurred when REORG is run. • labeled-duration-expression, SWITCHTIME labeled-duration-expression is added. |

Table 17. Alternate shared variable input data (continued)

| | | |
|-------|----------------------------------|--|
| USO43 | NEWMAXRO | <ul style="list-style-type: none"> N - NONE, specifies that when the specified SWITCHTIME is met, REORG proceeds to the last log iteration without taking log processing time into consideration. This option is the default. Integer, specifies the number of seconds. Valid values are 0 through 2147483647. |
| USO44 | RECLUSTER | <ul style="list-style-type: none"> Y - YES N - NO |
| USO45 | LISTPARTS | <ul style="list-style-type: none"> n - An integer representing the maximum number of data partitions to be reorganized at once. Valid values are integers 1 through 2147483647. |
| USO47 | PARALLEL | YES or an integer value between 0 and 32767. |
| USO50 | TABLE schema | Specifies the table owner for which STATISTICS information is to be gathered. |
| USO51 | TABLE name | Specifies the table name for which information is to be gathered. The table must belong to the specified table space. Multiple table names are not currently supported. Information may be gathered for all tables in the table space by specifying ALL for the table name and leaving the table owner blank. |
| USO52 | SAMPLE | Indicates the percentage of rows to sample when collecting non-indexed column statistics. Valid values are 1 through 100. The default is 25. |
| USO53 | COLUMN name | Specifies the columns for which column information is to be gathered. This option is valid only if a table name is specified. The utility accepts a maximum of 10 column names, but DB2 Admin does not validate this number. ALL means that statistics are to be gathered for all columns in the specified table name. |
| USO54 | COLGROUP name | Specifies that inline statistics will collect a cardinality value on the group of named columns. Multiple column groups are not currently supported. |
| USO55 | FREQVAL | <ul style="list-style-type: none"> Y - YES, collect frequency statistics N - NO, do not collect frequency statistics |
| USO56 | COUNT | Indicates the number of frequently occurring values to be collected from the specified column group. |
| USO57 | OCCUR | <ul style="list-style-type: none"> M - MOST, collect the most frequently occurring values B - BOTH, collect both the most and least frequently occurring values L - LEAST, collect the least frequently occurring values |
| USO58 | HISTOGRAM | <ul style="list-style-type: none"> Y - YES, gather histogram statistics from the specified column group N - NO, do not gather such statistics |
| USO59 | NUMQUANTILES for HISTOGRAM | Indicates the number of quantiles that the utility collects. |
| USO60 | INDEX(ALL) | <ul style="list-style-type: none"> Y - YES, gather information for all indexes on all tables in the table space N - NO, do not gather such information |
| USO61 | INDEX HISTOGRAM | <ul style="list-style-type: none"> Y - YES, gather histogram statistics for all indexes on all tables in the table space N - NO, do not gather such statistics |
| USO62 | NUMCOLS | The number of key columns that are to be concatenated when collecting histogram statistics from the specified index. |
| USO63 | NUMQUANTILES for INDEX HISTOGRAM | Indicates the number of quantiles that the utility collects. |

Table 17. Alternate shared variable input data (continued)

| | | |
|--------------------------|---------------------------|---|
| USORBALR | RBALRSN | Specifies the RBA and LRSN format in which the target object is to be left after a REORG. <ul style="list-style-type: none"> N - None No conversion B - Basic Convert to a basic format E - Extended Convert to extended format |
| RUNSTATS Utility Options | | |
| USR03 | SAMPLE | 1-100. This value is used as a parameter to SAMPLE. For example SAMPLE 37. |
| USR06 | FREQVAL COUNT | 1-65535. This value is used as a parameter to FREQVAL COUNT. For example FREQVAL COUNT 49. |
| USR07 | FREQVAL COUNT type | MOST, BEST, or LEAST. This value is used as a parameter to FREQVAL. For example: FREQVAL COUNT 50 LEAST. |
| USR10 | PART | 1-4096. This value is used as a parameter to PART. For example: PART 31. |
| USR11 | KEYCARD | Y or N. Use Y to specify KEYCARD or N to specify no keyword. |
| USR12 | NUMCOLS | A numeric value. This value is used as a parameter to NUMCOLS. For example: FREQVAL NUMCOLS 9 COUNT. |
| USR13 | NUMCOLS COUNT | 1 - 99999. This value is used as a parameter to COUNT. For example FREQVAL NUMCOLS 3 COUNT 7. |
| USR14 | NUMCOLS COUNT type | MOST, LEAST, or BOTH. This value is used as a parameter to COUNT. For example: FREQVAL NUMCOLS 3 COUNT 3 BOTH. |
| USR15 | SORTDEVT | A device type. This value is used as a parameter to SORTDEVT. For example: SORTDEVT <i>devt</i> . |
| USR16 | SORTNUM | 2-255. This value is used as a parameter to SORTNUM. For example SORTNUM 251. |
| USR17 | SHRLEVEL | R or C. Use R to specify SHRLEVEL REFERENCE or C to specify SHRLEVEL CHANGE. |
| USR18 | REPORT | Y or N. Use Y to specify REPORT YES or N to specify REPORT NO. |
| USR19 | UPDATE | A, P, S, or N. Use A to specify UPDATE ALL, P to specify UPDATE ACCESSPATH, S to specify UPDATE, or N to specify UPDATE NONE. |
| USR20 | HISTORY | A, P, S, or N. Use A to specify HISTORY ALL, P to specify HISTORY ACCESSPATH, S to specify HISTORY SPACE, or N to specify HISTORY NONE. |
| USR21 | FORCEROLLUP | Y or N. Use Y to specify FORCEROLLUP YES or N to specify FORCEROLLUP NO. |
| USR22 | NUMQUANTILES 1 | 1-100. This value is used as a parameter to NUMQUANTILES. For example HISTOGRAM NUMQUANTILES 8. |
| USR23 | NUMQUANTILES 2 | 1-100. This value is used as a parameter to NUMQUANTILES. For example HISTOGRAM NUMCOLS 3 NUMQUANTILES 61. |
| USR30 | PROFILE | USE or DELETE. Specify USE to specify USE PROFILE or DELETE to specify DELETE PROFILE. |
| USR31 | FROM EXISTING INCLUDE NPI | YES or NO. Use YES to specify INCLUDE NPI or NO to specify no keyword. |
| USR32 | TABLESAMPLE | AUTO or numeric literal between '.01' and '100'. This value is used as a parameter to TABLESAMPLE SYSTEM. For example TABLESAMPLE SYSTEM 7. |
| USR33 | REPEATABLE | A numeric value. This value is used as a parameter to REPEATABLE. For example REPEATABLE 65. |

Table 17. Alternate shared variable input data (continued)

| | | |
|------------------------|----------------------------|--|
| USR35 | SET PROFILE | SET or UPDATE. Use SET to specify SET PROFILE or UPDATE to specify UPDATE PROFILE. |
| USR36 | FROM EXISTING STATS | YES or NO. Use YES to specify FROM EXISTING STATS or NO to specify no keyword. |
| USR37 | HISTOGRAM NUMCOLS | A numeric value. This value is used as a parameter to HISTOGRAM NUMCOLS. For example HISTOGRAM NUMCOLS 89. |
| UNLOAD Utility Options | | |
| USU01 | FROMCOPY* | A data set name without quotation marks. This value is used as a parameter to FROMCOPY. For example FROMCOPY <i>my.dsn</i> . |
| USU02 | FROMVOLUME* | CATALOG or valid. This value is used as a parameter to FROMVOLUME. For example: FROMVOLUME <i>vol001</i> . |
| USU03 | FROMCOPYDDN* | DD name. This value is used as a parameter to FROMCOPYDDN. For example: FROMCOPYDDN <i>dd001</i> . |
| USU04 | ENCODINGScheme | E, A or U. Use E to specify EBCDIC, A to specify ASCII, or U to specify UNICODE. |
| USU05 | SBCS CCSID | A numeric value. This value is used as a parameter to CCSID. For example: CCSID(<i>n</i>). |
| USU06 | MIXED CCSID | A numeric value. This value is used as the second parameter to CCSID. For example: CCSID(1, <i>n</i>). |
| USU07 | DBCS CCSID | A numeric value. This value is used as the third parameter to CCSID. For example: CCSID(1, 2, <i>n</i>). |
| USU08 | NOSUBS | Y or N. Use Y to specify NOSUBS, or N to specify no keyword. |
| USU09 | NOPAD | Y or N. Use Y to specify NOPAD or N to specify no keyword. |
| USU10 | FLOAT | S or I. Use S to specify FLOAT S390 or I to specify FLOAT IEEE. |
| USU11 | MAXERR | A numeric value. This value is used as a parameter to MAXERR. For example MAXERR 47. |
| USU12 | SHRLEVEL | 1, 2 or 3. Use 1 to specify SHRLEVEL CHANGE ISOLATION CS, 2 to specify SHRLEVEL CHANGE ISOLATION UR, or 3 to specify SHRLEVEL REFERENCE. |
| USU13 | DELIMITED | Y or N. Use Y to specify DELIMITED or N to specify no keyword. |
| USU17 | HEADER | O, N or C. Use O to specify HEADER OBID, N to specify HEADER NONE, or C to specify HEADER CONST #. |
| USU18 | CONST | A character or X'hex string'. This value is used as a parameter to CONST. For example: HEADER CONST #. |
| USU19 | SAMPLE | A percent, where 0 < x <= 100. This value is used as a parameter to SAMPLE. For example SAMPLE 22. |
| USU20 | LIMIT | An integer, 0 - 2147483647. This value is used as a parameter to LIMIT. For example: LIMIT 20. |
| USU21 | SKIP LOCKED DATA | YES or NO. Use YES to specify SKIP LOCKED DATA or NO to specify no keyword. |
| USU22 | This variable is not used. | |
| USU23 | CLONE | YES or NO. Use YES to specify CLONE or NO to specify no keyword. |
| USU24 | IMPLICIT TIMEZONE | +NN:NN, -NN:NN. This value is used as a parameter to IMPLICIT_TZ. For example: IMPLICIT_TZ +7. |
| USU25 | SPANNED | YES or NO. This value is used as a parameter for SPANNED. For example: SPANNED YES. |
| USU27 | PARALLEL | YES or an integer value between 0 and 32767. |
| USURND | DECFLOAT_ROUNDMODE | ROUND_CEILING, ROUND_DOWN, ROUND_FLOOR, ROUND_HALF_DOWN, ROUND_HALF_EVEN, ROUND_HALF_UP, or ROUND_UP. This value is used as a parameter to DECFLOAT_ROUNDMODE. For example: DECFLOAT_ROUNDMODE ROUND_UP. |
| USUUF1 | FORMAT INTERNAL | Y or N. Use Y to specify FORMAT INTERNAL or N to specify no keyword. |

Table 17. Alternate shared variable input data (continued)

| | | |
|----------------------|------------------------|---|
| USULIC | LAST IC | LAST, BEFORE, or AFTER. This value is used as a parameter to FROM. Use LAST to specify FROM LAST_IC, BEFORE to specify FROM BEFORE_IC, or AFTER to specify FROM AFTER_IC. |
| USUICD | IC date | A date, YYYY/MM/DD. This value is used as a parameter to ICDATE. For example: FROM LAST_IC ICDATE 2013/08/04. |
| USUICT | IC time | Time, HH:MM:SS, used as a parameter to ICDATE. For example FROM LAST_IC ICTIME 12:04:00. |
| | | |
| LOAD Utility Options | | |
| UTC01 | UTILITY ID | A name. This value is used for UID parameter. For example: //LOAD1 EXEC DSNUPROC,SYSTEM=DSNA,UID=' P5V01'. |
| UTC02 | DSNAME | A data set name. This value is used as the SYSREC data set name. For example: //DSNUPROC.SYSREC DD DISP=SHR,DSN= my.dsn. |
| UTC03 | DSNAME into-table-spec | A data set name. The data set contains LOAD ... INTO TABLE ... statements. |
| UTC04 | RESUME | YES or NO. This value is used as a parameter to RESUME. For example: RESUME YES. |
| UTC05 | SHRLEVEL | NONE or CHANGE. This value is used as a parameter to SHRLEVEL. For example: SHRLEVEL CHANGE. |
| UTC06 | REPLACE | YES or NO. Use YES to specify REPLACE or NO to specify no keyword. |
| UTC07 | COPYDDN1 | A name. This value is used as a parameter to COPYDDN. For example: COPYDDN(name). |
| UTC08 | COPYDDN2 | A name. This value is used as a parameter to COPYDDN. For example: COPYDDN(, name). |
| UTC09 | RECOVERYDDN1 | A name. This value is used as a parameter to RECOVERYDDN. For example: RECOVERYDDN(name). |
| UTC10 | RECOVERYDDN2 | A name. This value is used as a parameter to RECOVERYDDN. For example: RECOVERYDDN(ddn1, name). |
| | | |
| UTC12 | SAMPLE* | An integer, 1- 100. This value is used as a parameter to SAMPLE. For example: SAMPLE 48. |
| UTC13 | INDEX ALL* | YES or NO. Use YES to specify INDEX(ALL) or NO to specify no keyword. |
| UTC14 | REPORT* | YES or NO. Use YES to specify REPORT YES or NO to specify no keyword. |
| UTC15 | UPDATE* | A, P, S, or N. Use A to specify UPDATE ALL, P to specify UPDATE ACCESSPATH, S to specify UPDATE SPACE, or N to specify UPDATE NONE. |
| UTC16 | KEEPDICTIONARY | YES or NO. Use YES to specify KEEPDICTIONARY or NO to specify no keyword. |
| UTC17 | REUSE | YES or NO. Use YES to specify REUSE or NO to specify no keyword. |
| UTC18 | LOG | YES, NO, or NOC. Use YES to specify LOG YES, NO to specify LOG NO or NOC to specify LOG NO NOCOPYPEND. |
| UTC19 | WORKDDN1 | A name. This value is used as a parameter to WORKDDN. For example: WORKDDN(name). |
| UTC20 | WORKDDN2 | A name. This value is used as a parameter to WORKDDN. For example: WORKDDN(, name). |
| UTC21 | SORTKEYS | An integer, 0 - 2147483647. This value is used as a parameter to SORTKEYS. For example: SORTKEYS 39. |
| UTC22 | ENFORCE | YES or NO. Use YES to specify ENFORCE CONSTRAINTS or NO to specify ENFORCE NO. |
| UTC23 | SORTDEVT | A device type. This value is used as a parameter to SORTDEVT. For example: SORTDEVT SYSALLDA. |

Table 17. Alternate shared variable input data (continued)

| | | |
|----------|--|---|
| UTC24 | SORTNUM | 1- 255. This value is used as a parameter to SORTNUM. For example: SORTNUM 12. |
| UTC25 | SORTWK | 0, 1, 2, 3, or 4. This parameter determines how many sort work DD statements are allocated. DD statements for SORTWK01, SORTWK02, SORTWK03, and SORTWK04 may be added. |
| UTC26 | how unloaded | U or R. Use U to specify WHEN (00001:00002 = 'X&XOBID') or R to specify WHEN (00004:00005 = 'X&XOBID'). |
| UTC27 | DECFLOAT ROUNDING | Use &UTCRRND instead. |
| UTC28 | IMPLICIT_TZ | +NN:NN, -NN:NN. This value is used as a parameter to IMPLICIT_TZ. For example: IMPLICIT_TZ +08. |
| UTC29 | FLASHCOPY | Y, N or C. Use Y to specify FLASHCOPY YES, N to specify FLASHCOPY NO, or C to specify FLASHCOPY CONSISTENT. |
| UTC30 | PRESORTED | YES or NO. This value is used as a parameter to PRESORTED. For example: PRESORTED YES. |
| UTC31 | PARALLEL (DB2 V11 and above) | YES or an integer value between 0 and 32767. |
| UTC40 | Table schema* | Blank. Use Blank to not specify STATISTICS TABLE(table-name) because it is not supported in the compare process. |
| UTC41 | Table name* | Y, ALL or Blank. Use ALL to specify STATISTICS TABLE(ALL) or Blank to specify no keyword. |
| UTC54 | DISCARDS | 0 - 2147483647. This value is used as a parameter to DISCARDS. For example, DISCARDS 12. |
| UTCRRND | DECFLOAT ROUNDING | ROUND_CEILING, ROUND_DOWN, ROUND_FLOOR, ROUND_HALF_DOWN, ROUND_HALF_EVEN, ROUND_HALF_UP, ROUND_UP. This value is used as a parameter to DECFLOAT_ROUNDMODE. For example: DECFLOAT_ROUNDMODE ROUND_UP. |
| UTNCOPY1 | Name of data set for copy (ADB25TU) | A template name. This value is used as the first parameter to the COPYDDN keyword. For example: COPYDDN(<i>mytemp1</i>). |
| UTNCOPY2 | Name of data set for copy (ADB25TU) | A template name. This value is used as the second parameter to the COPYDDN keyword. For example: COPYDDN(<i>mytemp1</i> , <i>mytemp2</i>). |
| UTNDISC1 | Template discard data set name | A template name. This value is used as the parameter to the DISCARDDDN keyword. For example: DISCARDDDN(<i>mytemp3</i>). |
| UTNERR | Template error data set name (ADB25TU) | A template name. This value is used as the parameter to the ERRDDN keyword. For example: ERRDDN(<i>mytemp4</i>). |
| UTNFCOPY | Name of utility data set for system FCCOPY (ADB25TU) | A template name. This value is used as the parameter to the FCCOPYDDN keyword. For example: FCCOPYDDN(<i>mytemp5</i>). |
| UTNFILTR | Name of utility data set for system filter (ADB25TU) | A template name. This value is used as the parameter to the FILTERDDN keyword. For example: FILTERDDN(<i>mytemp6</i>). |
| UTNMAPDD | Name of utility data set for system map (ADB25TU) | A template name. This value is used as the parameter to the MAPDDN keyword. For example: MAPDDN(<i>mytemp7</i>). |
| UTNPUNCH | Name of utility data set for system punch (ADB25TU) | A template name. This value is used as the parameter to the PUNCHDDN keyword. For example: PUNCHDDN(<i>mytemp8</i>). |
| UTNRECV1 | Name of recovery data set (ADB25TU) | A template name. This value is used as the first parameter to the RECOVERYDDN keyword. For example: RECOVERYDDN(<i>mytemp9</i>). |
| UTNRECV2 | Name of recovery data set (ADB25TU) | A template name. This value is used as the second parameter to the RECOVERYDDN keyword. For example: RECOVERYDDN(<i>mytemp9</i> , <i>mytempA</i>). |
| UTNUNLDD | Name of utility data set for unload (ADB25TU) | A template name. This value is used as the parameter to the UNLDDN keyword. For example: UNLDDN(<i>mytempB</i>). |

Table 17. Alternate shared variable input data (continued)

| | | |
|------------------------------|---|--|
| UTNWORK1 | Name of utility data set for work (ADB25TU) | A template name. This value is used as the first parameter to the WORKDDN keyword. For example: WORKDDN(<i>mytempc</i>). |
| UTNWORK2 | Name of utility data set for work (ADB25TU) | A template name. This value is used as the second parameter to the WORKDDN keyword. For example: WORKDDN(<i>mytempC</i> , <i>mytempD</i>). |
| UTUCOPY1 | Use indicator for copy template (ADB25TU) | / or blank. Specify / to use, or blank to not use &UTNCOPY1 |
| UTUCOPY2 | Use indicator for copy template (ADB25TU) | / or blank. Specify / to use, or blank to not use &UTNCOPY2 |
| UTUDISC1 | Use indicator for template discard name | / or blank. Specify / to use, or blank to not use &UTNDISC1 |
| UTUERR | Use indicator for ERROR template (ADB25TU) | / or blank. Specify / to use, or blank to not use &UTNERR |
| UTUFCOPY | Use indicator for FCCOPY template (ADB25TU) | / or blank. Specify / to use, or blank to not use &UTNFCOPY |
| UTUFILTR | Use indicator for filter template (ADB25TU) | / or blank. Specify / to use, or blank to not use &UTNFILTR |
| UTUMAPDD | Use indicator for Map template (ADB25TU) | / or blank. Specify / to use, or blank to not use &UTNMAPDD |
| UTUPUNCH | Use indicator for punch template (ADB25TU) | / or blank. Specify / to use, or blank to not use &UTNPUNCH |
| UTURECV1 | Use indicator for recovery template (ADB25TU) | / or blank. Specify / to use, or blank to not use &UTNRECV1 |
| UTURECV2 | Use indicator for recovery template (ADB25TU) | / or blank. Specify / to use, or blank to not use &UTNRECV2 |
| UTUUNLDD | Use indicator for unload template (ADB25TU) | / or blank. Specify / to use, or blank to not use &UTNUNLDD |
| UTUWORK1 | Use indicator for work1 template (ADB25TU) | /. This value must be set to "/" to use &UTNWORK1 |
| UTUWORK2 | Use indicator for work2 template (ADB25TU) | / or blank. Specify / to use, or blank to not use &UTNWORK2 |
| REORG INDEX utility options: | | |
| UXO01 | REUSE | Y or N. Use Y to specify REUSE or N to specify no keyword. |
| UXO02 | SHRLEVEL | R or C. Use R to specify SHRLEVEL REFERENCE or C to specify SHRLEVEL CHANGE. |
| UXO03 | LEAFDISTLIMIT | Valid values are 0 through 2147483647. |
| UXO04 | REPORTONLY | Y or N. Use Y to specify REPORTONLY or N to specify no keyword. |
| UXO05 | UNLOAD | C, P, O, or E. Use C to specify UNLOAD CONTINUE, P to specify UNLOAD PAUSE, O to specify UNLOAD ONLY, or E to specify UNLOAD EXTERNAL. |
| UXO06 | PREFORMAT | Y or N. Use Y to specify PREFORMAT or N to specify no keyword. |
| UXO07 | DEADLINE | N, timestamp, or labeled duration expression. Use N to specify DEADLINE NONE together with a timestamp that is used as a parameter to DEADLINE. For example: DEADLINE 13:15:01. An example of a labeled duration expression is CURRENT_DATE +3 DAYS. |
| UXO08 | DRAIN_WAIT | 1-1800. This value is used as a parameter to DRAIN_WAIT. |
| UXO09 | RETRY | 0-255. This value is used as a parameter to RETRY. For example: RETRY 8. |
| UXO10 | RETRY_DELAY | 1-1800. This value is used as a parameter to RETRY_DELAY. For example RETRY_DELAY 17. |
| UXO11 | FASTSWITCH | Y or N. Use Y to specify FASTSWITCH YES or N to specify FASTSWITCH NO. |
| UXO12 | MAXRO | D or numeric value. Use D to specify MAXRO DEFER or numeric value to specify MAXRO &uso24. |
| UXO13 | DRAIN | W or A. Use W to specify DRAIN WRITERS or A to specify DRAIN ALL. |

Table 17. Alternate shared variable input data (continued)

| | | |
|-------|--------------|---|
| UXO14 | LONGLOG | C, T, or D. Use C to specify LONGLOG CONTINUE, T to specify LONGLOG TERM, or D to specify LONGLOG DRAIN. |
| UXO15 | DELAY | A numeric value. This value is used as a parameter to DELAY. For example DELAY 17. |
| UXO16 | TIMEOUT | A or T. Use A to specify TIMEOUT ABEND or T to specify TIMEOUT TERM. |
| UXO17 | STATISTICS | Y or N. Use Y to specify STATISTICS TABLE (ALL) or N to specify no keyword. |
| UXO18 | REPORT | Y or N. Use Y to specify REPORT YES or N to specify REPORT NO. |
| UXO19 | KEYCARD | Y or N. Use Y to specify KEYCARD or N to specify no keyword. |
| UXO20 | FREQVAL | Indicates that frequency statistics are to be gathered from the specified column group. <ul style="list-style-type: none"> • Y - YES, collect frequency statistics • N - NO, do not collect frequency statistics |
| UXO21 | NUMCOLS | A numeric value. This value is used as a parameter to NUMCOLS. For example: FREQVAL NUMCOLS 9 COUNT. |
| UXO22 | COUNT | Indicates the number of frequently occurring values to be collected from the specified column group. |
| UXO23 | UPDATE | A, P, S, or N. Use A to specify UPDATE ALL, P to specify UPDATE ACCESSPATH, S to specify UPDATE, or N to specify UPDATE NONE. |
| UXO24 | HISTORY | A, P, S, or N. Use A to specify HISTORY ALL, P to specify HISTORY ACCESSPATH, S to specify HISTORY SPACE, or N to specify HISTORY NONE. |
| UXO25 | FORCEROLLUP | Y or N. Use Y to specify FORCEROLLUP YES or N to specify FORCEROLLUP NO. |
| UXO26 | SORTDEVT | A device type. This value is used as a parameter to SORTDEVT. For example: SORTDEVT SYSALLDA. |
| UXO27 | SORTNUM | 1- 255. This value is used as a parameter to SORTNUM. For example: SORTNUM 12. |
| UXO28 | CLONE | Y or N. Use Y to specify CLONE or N to specify no keyword. |
| UXO29 | FLASHCOPY | Y, C, or N. Use Y to specify FLASHCOPY YES, C to specify FLASHCOPY CONSISTENT, or N to specify FLASHCOPY NO. |
| UXO30 | HISTOGRAM | <ul style="list-style-type: none"> • Y - YES, gather histogram statistics from the specified column group • N - NO, do not gather such statistics |
| UXO31 | NUMCOLS | The number of key columns that are to be concatenated when collecting histogram statistics from the specified index. |
| UXO32 | NUMQUANTILES | Indicates the number of quantiles that the utility collects. |
| UXO40 | LOGRANGES | <ul style="list-style-type: none"> • Y - Yes, REORG uses SYSLGRNX information for the LOG phase whenever possible. This option is the default behavior. • N - NO, REORG does not use SYSLGRNX information for the LOG phase. |
| UXO42 | SWITCHTIME | <ul style="list-style-type: none"> • N - NONE, does not specify a time for the final log iteration of the LOG phase. This option is the default behavior. Specifies the time that the final log iteration of the LOG phase is to begin. This time must not have already occurred when REORG is run. • labeled-duration-expression, SWITCHTIME labeled-duration-expression is added. |

Table 17. Alternate shared variable input data (continued)

| | | |
|----------|----------|--|
| UXO43 | NEWMAXRO | <ul style="list-style-type: none"> • N - NONE, specifies that when the specified SWITCHTIME is met, REORG proceeds to the last log iteration without taking log processing time into consideration. This option is the default. • Integer, specifies the number of seconds. Valid values are 0 through 2147483647. |
| UXORBALR | RBALRSN | <p>Specifies the RBA and LRSN format in which the target object is to be left after a REORG.</p> <ul style="list-style-type: none"> • N - None <p style="padding-left: 20px;">No conversion</p> <ul style="list-style-type: none"> • B - Basic <p style="padding-left: 20px;">Convert to a basic format</p> <ul style="list-style-type: none"> • E - Extended <p style="padding-left: 20px;">Convert to extended format</p> |

Refer to *IBM DB2 Administration Tool for z/OS User's Guide and Reference* for additional information about utilities.

Creating user-defined templates

You can create a data set template in DB2 Object Comparison Tool to save DB2 Admin Change Management batch parameter variables. After you define a data set with DB2 TEMPLATE statements, you can reuse these template statements in apply jobs.

About this task

Without a reusable template, the settings of each new apply job that you run overwrite the settings of your previous apply job. To create a reusable template, you must save the Change Management batch variables in USRTEMPL DD. USRTEMPL and ADB25TU templates can be used at the same time. USRTEMPL templates take precedence over 25TU templates.

Procedure

1. Create a data set and name it ADBPRE.USRTEMPL.
 - a. Define the logical record length of 80.
 - b. Enter the prefix value ADBPRE in panel GOC5, ADBPALT, or ADB2C11A, depending on the method you are using to run the apply job.
2. Add the Change Management batch parameters and variables for the templates that you want to use. In the USRTEMPL data set, you must set the parameters before adding the templates. You can store the following Change Management batch parameters in the USRTEMPL data set:
 - UTIL_TEMPLATE_DISCARDNN_NAME
 - UTIL_TEMPLATE_DISCARDNN_USE
 - UTIL_CLONE_TEMPLATE_DISCARDNN_NAME
 - UTIL_CLONE_TEMPLATE_DISCARDNN_USE
 - UTIL_TEMPLATE_DISCARDNC_NAME
 - UTIL_TEMPLATE_DISCARDNC_USE
 - UTIL_CLONE_TEMPLATE_DISCARDNC_NAME

- UTIL_CLONE_TEMPLATE_DISCARDNC_USE
- UTIL_TEMPLATE_UNLOAD_PUNCHDDN_NAME
- UTIL_TEMPLATE_UNLOAD_PUNCHDDN_USE
- UTIL_CLONE_TEMPLATE_UNLOAD_PUNCHDDN_NAME
- UTIL_CLONE_TEMPLATE_UNLOAD_PUNCHDDN_USE
- UTIL_TEMPLATE_UNLOAD_PUNCHDDNC_NAME
- UTIL_TEMPLATE_UNLOAD_PUNCHDDNC_USE
- UTIL_CLONE_TEMPLATE_UNLOAD_PUNCHDDNC_NAME
- UTIL_CLONE_TEMPLATE_UNLOAD_PUNCHDDNC_USE
- UTIL_TEMPLATE_UNLOAD_UNLDDN_NAME
- UTIL_TEMPLATE_UNLOAD_UNLDDN_USE
- UTIL_TEMPLATE_UNLOAD_UNLDDNC_NAME
- UTIL_TEMPLATE_UNLOAD_UNLDDNC_USE
- UTIL_CLONE_TEMPLATE_UNLOAD_UNLDDN_NAME
- UTIL_CLONE_TEMPLATE_UNLOAD_UNLDDNC_NAME
- UTIL_TEMPLATE_COPYDDN1_NAME
- UTIL_CLONE_TEMPLATE_COPYDDN1_NAME
- UTIL_TEMPLATE_COPYDDN2_NAME
- UTIL_CLONE_TEMPLATE_COPYDDN2_NAME
- UTIL_TEMPLATE_ERRDDN_NAME
- UTIL_CLONE_TEMPLATE_ERRDDN_NAME
- UTIL_TEMPLATE_FCCOPYDDN_NAME
- UTIL_CLONE_TEMPLATE_FCCOPYDDN_NAME
- UTIL_TEMPLATE_LOBCOL_NAME
- UTIL_CLONE_TEMPLATE_LOBCOL_NAME
- UTIL_TEMPLATE_MAPDDN_NAME
- UTIL_CLONE_TEMPLATE_MAPDDN_NAME
- UTIL_TEMPLATE_PUNCHDDN_NAME
- UTIL_CLONE_TEMPLATE_PUNCHDDN_NAME
- UTIL_TEMPLATE_RECOVERYDDN1_NAME
- UTIL_CLONE_TEMPLATE_RECOVERYDDN1_NAME
- UTIL_TEMPLATE_RECOVERYDDN2_NAME
- UTIL_CLONE_TEMPLATE_RECOVERYDDN2_NAME
- UTIL_TEMPLATE_UNLDDN_NAME
- UTIL_CLONE_TEMPLATE_UNLDDN_NAME
- UTIL_TEMPLATE_WORKDDN1_NAME
- UTIL_CLONE_TEMPLATE_WORKDDN1_NAME
- UTIL_TEMPLATE_WORKDDN2_NAME
- UTIL_CLONE_TEMPLATE_WORKDDN2_NAME
- UTIL_CLONE_TEMPLATE_WORKDDN2_NAME
- UTIL_TEMPLATE_XMLCOL_NAME
- UTIL_CLONE_TEMPLATE_XMLCOL_NAME
- UTIL_TEMPLATE_COPYDDN1_USE
- UTIL_CLONE_TEMPLATE_COPYDDN1_USE
- UTIL_TEMPLATE_COPYDDN2_USE

- UTIL_CLONE_TEMPLATE_COPYDDN2_USE
- UTIL_TEMPLATE_ERRDDN_USE
- UTIL_CLONE_TEMPLATE_ERRDDN_USE
- UTIL_TEMPLATE_FCCOPYDDN_USE
- UTIL_CLONE_TEMPLATE_FCCOPYDDN_USE
- UTIL_TEMPLATE_LOBCOL_USE
- UTIL_CLONE_TEMPLATE_LOBCOL_USE
- UTIL_TEMPLATE_MAPDDN_USE
- UTIL_CLONE_TEMPLATE_MAPDDN_USE
- UTIL_TEMPLATE_PUNCHDDN_USE
- UTIL_CLONE_TEMPLATE_PUNCHDDN_USE
- UTIL_TEMPLATE_RECOVERYDDN1_USE
- UTIL_CLONE_TEMPLATE_RECOVERYDDN1_USE
- UTIL_TEMPLATE_RECOVERYDDN2_USE
- UTIL_CLONE_TEMPLATE_RECOVERYDDN2_USE
- UTIL_TEMPLATE_UNLDDN_USE
- UTIL_CLONE_TEMPLATE_UNLDDN_USE
- UTIL_TEMPLATE_WORKDDN1_USE
- UTIL_CLONE_TEMPLATE_WORKDDN1_USE
- UTIL_TEMPLATE_WORKDDN2_USE
- UTIL_CLONE_TEMPLATE_WORKDDN2_USE
- UTIL_TEMPLATE_XMLCOL_USE
- UTIL_CLONE_TEMPLATE_XMLCOL_USE

3. Add the template by typing one of the following formats in the data set:

- DB2 template format


```

      TEMPLATE template_name
      DSN dsn_definition
      template_details
      
```
- XML template format


```

      <TEMPLATE>
      <NAME>
      template_name
      </NAME>
      <DSN>
      dsn_definition
      </DSN>
      <OTHER>
      template_details
      </OTHER>
      </TEMPLATE>
      
```

What to do next

You can now use USRTEMPL data sets to run different apply jobs without losing the template settings for each job due to overwrites. Reusable templates are useful if you are frequently running more than one apply job.

Related tasks:

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.

| **Related reference:**

| Parameter definitions: Change Management batch interface

| The following Change Management batch interface parameters can be used to
| control Change Management actions and settings.

Chapter 14. Comparing 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.

When you use DB2 Object Comparison Tool to compare a large number of objects, follow these recommendations:

- Specify a large region size on the job card, for example 256 MB, to ensure that the batch job can get sufficient virtual storage.
- Ensure that your batch jobs can get sufficient CPU time. When you compare 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 300 (CPU minutes).
- Ensure that the data sets for the version file output are large enough to contain the data for the objects. If the data sets are not large enough, Step 1 or Step 2 of the compare batch job can terminate with a x37 abend. To prevent this storage problem, modify the JCL before submitting the job to use `PACE=(CYL,(10,100))` for the following data sets:
 - CAT (in two places)
 - SRCSIN
 - SRCSOUT
 - TGTSOUT
- To avoid data set extension failures caused by referback, allocate the data set up front.

The following JCL shows how the JCL should look before and after you modify it:

Before

```
//CAT      DD DSN= ....
//          DISP=(NEW,CATLG,DELETE),
//          DCB=(LRECL=16384,RECFM=VB,BLKSIZE=27998),
//          SPACE=(CYL,(10,10),RLSE),
//          UNIT=SYSDA
(in two places)
...
//SRCSIN   DD DSN=&SRCSIN,DISP=(,DELETE),
//          DCB=(LRECL=16384,RECFM=VB,BLKSIZE=27998),
//          SPACE=(CYL,(10,20),RLSE),
//          UNIT=SYSALLDA
//SRCSOUT  DD DSN=&SRCSOUT,DISP=(,DELETE),
//          DCB=(LRECL=16384,RECFM=VB,BLKSIZE=27998),
//          SPACE=(CYL,(10,20),RLSE),
//          UNIT=SYSALLDA
//TGTSIN   DD DISP=SHR,
//          DSN= ....
//TGTSOUT  DD DSN=&TGTSOUT,DISP=(,DELETE),
//          DCB=(LRECL=16384,RECFM=VB,BLKSIZE=27998),
//          SPACE=(CYL,(10,20),RLSE),
//          UNIT=SYSALLDA
```

After

```
//CAT      DD DSN= ....
//          DISP=(NEW,CATLG,DELETE),
//          DCB=(LRECL=16384,RECFM=VB,BLKSIZE=27998),
//          SPACE=(CYL,(10,100),RLSE),
//          UNIT=SYSDA
(in two places)
```

```
...
//SRCSIN DD DSN=&SRCSIN,DISP=(,DELETE),
//        DCB=(LRECL=16384,RECFM=VB,BLKSIZE=27998),
//        SPACE=(CYL,(10,100),RLSE),
//        UNIT=SYSALLDA
//SRCSOUT DD DSN=&SRCSOUT,DISP=(,DELETE),
//        DCB=(LRECL=16384,RECFM=VB,BLKSIZE=27998),
//        SPACE=(CYL,(10,100),RLSE),
//        UNIT=SYSALLDA
//TGTSIN DD DISP=SHR,
//        DSN= ....
//TGTSOUT DD DSN==&TGTSOUT,DISP=(,DELETE),
//        DCB=(LRECL=16384,RECFM=VB,BLKSIZE=27998),
//        SPACE=(CYL,(10,100),RLSE),
//        UNIT=SYSALLDA
```

Chapter 15. Reference

Use the DB2 Object Comparison Tool reference information when you need more information about the customization of the components. You can find details about the tasks, steps, and parameters that are displayed on the Component Parameters panel, and information about the generated jobs.

DB2 Object Comparison Tool parameters

The following table shows the parameters for DB2 Object Comparison Tool that you can specify by using Tools Customizer. These parameters are displayed on the Product Parameters panel.

Table 18. DB2 Object Comparison Tool parameters

| Parameter name | Parameter description |
|----------------------|---|
| CCQ_GOC_ADB_HLQ | The high-level qualifier of the DB2 Admin Tool data sets, one of which contains the EXEC (ADBFB2VB) which performs the conversion. |
| CCQ_GOC_HLQ | The high-level qualifier of the DB2 Object Comparison Tool data sets for the product customization step. |
| CCQ_GOC_FB2VB_VLSRNM | The volume_serial used for copying fixed-block libraries to variable-blocked libraries. This is not necessary for SMS (System Managed Storage). |
| CCQ_GOC_FB2VB_DASD | The UNIT name used for copying fixed-block libraries to variable-blocked libraries. This is not necessary for SMS(System Managed Storage). |

Customization jobs generated by Tools Customizer

Tools Customizer generates customization jobs based on the tasks and steps that you select.

The following table shows the relationship between the tasks and steps that you select, and the member that contains the jobs that Tools Customizer generates.

Table 19. List of customization jobs that Tools Customizer can generate for DB2 Object Comparison Tool

| Tasks | Steps | Template name | Template type |
|---|---|---------------|---------------|
| Create the VB CLIST and EXEC libraries. | Create the VB libraries. | GOCFB2VB | perhlq |
| Sample JCL step for CM Batch | Create the CM Batch interface job step. | GOCCM2 | perhlq |

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 Object Comparison Tool 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 Object Comparison Tool. If the product to be customized requires DB2 entries, you can customize DB2 Object Comparison Tool only on DB2 entries that are in the associated list. When you customize DB2 Object Comparison Tool, 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 Object Comparison Tool.

Master list

The list of all DB2 entries that are defined but are not associated with DB2 Object Comparison Tool. 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 Object Comparison Tool, 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 Object Comparison Tool, 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 Object Comparison Tool. These parameters are defined by DB2 Object Comparison Tool and are stored in a data member that is defined by DB2 Object Comparison Tool.

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 Object Comparison Tool, all of the required parameters must be defined.

If required parameters for the the product 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 and DB2 parameters for the DB2 entries on which DB2 Object Comparison Tool will be customized.

The following table shows the meaning of the status types. Each status is defined differently for each type of parameter.

Table 20. 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 42

You can create new DB2 entries and associate them with DB2 Object Comparison Tool.

“Copying DB2 entries” on page 50

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

“Removing DB2 entries” on page 52

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 Object Comparison Tool with Tools Customizer. These data sets are supplied by DB2 Object Comparison Tool, supplied by Tools Customizer, or allocated by Tools Customizer.

DB2 Object Comparison Tool 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*.SGOCDENU, 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 Object Comparison Tool Discover EXEC. When you customize DB2 Object Comparison Tool, 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 Object Comparison Tool, the lowest-level qualifier is SGOCDENU. 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 Object Comparison Tool. 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 Object Comparison Tool Discover EXEC. The DB2 Object Comparison Tool Discover EXEC retrieves the metadata and values for the parameters from a previous customization of DB2 Object Comparison Tool.

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 Object Comparison Tool.

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 Object Comparison Tool, 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*\$.*xyzvm*, 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
- *xyzvm* 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\$.XYZ410.

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 21. 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 |
| 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 16. Troubleshooting and messages

Use this information to diagnose and correct problems that you might experience when you customize DB2 Object Comparison Tool.

Important: This section includes only the messages that you might encounter during the Tools Customizer customization process. For the complete list of messages, see the DB2 Administration Tool for z/OS user's guide.

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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 ISPFTRC command, and press Enter.
3. Issue the Tools Customizer command that issues the error.
4. Specify the ISPFTRC 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 DB2 member 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. Contact IBM Software Support.

information” on page 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 *mininum_number* times.

Explanation: The specified attribute did not occur enough times.

System action: Processing stops.

User response: See “Gathering diagnostic information” on page 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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.SGOCDENU*.

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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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_LPAR_, or CCQ_PRD_.

System action: Processing continues.

User response: See “Gathering diagnostic information” on page 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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 241. 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.SGOCDENU*.

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.

CCQQ002E • CCQS002W

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.SxxxSAMP.

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.SxxxSAMP, 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 *nm*, where *nm* 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 *nm*, where *nm*
| 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.

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 Object Comparison Tool.

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.

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.

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.

Programming interface information

This publication documents information that is NOT intended to be used as Programming Interfaces of DB2 Object Comparison Tool.

Trademarks

IBM, the IBM logo, and [ibm.com](http://www.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

A

- accessibility
 - overview 19
- accessing OCT from DB2 Administration Tool 59
- ADB25TU 113
- ADB27WLD 113
- ADB2C22 90, 95
- ADB2CL2 90
- ADB2CRO 111
- ADB2GEN program 153
- ADB2SD 139
- ADB2SDS 138
- ADB2UCUS skeleton
 - configuration
 - explained 58
 - restrictions 58
 - SET statements 58
 - configuration variables 58
- ADB2UOPS 113
- ADB2UPA 113
- ADB2W1R 136
- ADB2WLDA 113
- ADBL CLIST 55
- ALTER apply job 113
- ALTER DATABASE 147
- ALTER FUNCTION 147
- alter identity column attributes 158
- ALTER INDEX 147
- ALTER PROCEDURE 147
- ALTER SEQUENCE 147
- alternate forms of syntax 6
- alternate input data set
 - creating 211
 - listing current values 211
 - requirements 211
- alternate input to the generate apply job 211
- alternate statements 6
- apply jobs
 - ALTER 113
 - CREATE 113
 - DB2 libraries 61
 - DROP 113
 - generating a single apply job 113
 - generating multiple apply jobs 113
 - REBIND 113
 - RELOAD 113
 - REORG 113
 - START 113
 - STOP 113
 - target system 61
 - UNLOAD 113
 - utilities 113
- applying masks 187
- authorization ID masks 187
- authorization switching 132

B

- batch DB2 catalog extraction programs
 - sample output 153
- batch DDL file extraction programs
 - sample output 150
- batch job
 - compare exclude 168
- Batch Job Utility Parameters panel (ADB2UPA) 113
- batch jobs
 - creating 61
 - creating version file from source 62
 - parameters 113
 - specifying options 105, 132
- buffer pool analysis
 - overview 2

C

- catalog searches 75, 77
- change
 - updating targets 135
- change dialog location 140
- change management
 - updating targets 135
- Change Management
 - CM - Register Options panel (ADB2CRO) 111
 - generating a compare batch job 111
 - registering changes 111
 - specifying compare masks 90
 - specifying ignore fields 95
- changing display options 38
- changing reporting options
 - changed objects 113
 - ignore fields 113
 - system-generated ignore fields 113
 - object count reports 113
 - summary reports 113
 - translation masks 113
- CHECK DATA, run after RELOAD 113
- CLIST, ADBL, preparing 55
- COMMENT ON 147
- COMMIT 147
- Compare Add Databases panel (GOC1DA) 69
- Compare Add Databases panel (GOC1DD) 69
- Compare Add Schema panel (GOC1CA) 77
- Compare Add Schema panel (GOC1CD) 77
- Compare Add Table Spaces panel (GOC1SA) 72
- Compare Add Table Spaces panel (GOC1SD) 72
- Compare Add Tables panel (GOC1TA) 75

- Compare Add Tables panel (GOC1TD) 75
- compare exclude
 - batch job 168
- compare ignore changes
 - specifying 168
- compare jobs
 - creating automatically 61, 62
 - specifying options 105
 - specifying options (multi-target import) 132
- compare masks
 - applying 187
 - authorization ID 187
 - authorization ID masks 187
 - editing
 - editing masks stored in the Change Management database 90
 - examples 187
 - hierarchy 192
 - name masks 187
 - specifying 90, 167
 - specifying in the Change Management database 90
 - syntax 187
 - valid names 192
- compare options
 - run SQL ID 113
 - suppress adding columns 113
 - suppress DROP of columns 113
 - suppress DROP of objects 113
- compare process
 - exceptions 140
 - exclude 140
 - include 140
- compare results
 - managing 107
 - saving 107
- comparing functions 158
- comparing multiple objects 62, 140
- comparing object types 80
- comparing partitioned tables 158
- comparing renamed objects 158
- comparing table columns 158
- comparing triggers 158
- comparing version files 158
- comparing views 158
- components 5
- condition codes
 - defined 169
 - documentation for support 169
 - user action 169
- configuration
 - ADB2UCUS skeleton 58
 - variables 58
- configuration parameters
 - creating 59
 - location 59
- configuring the main menu 59
- constraint names 158

- cookie policy 301, 303
- Copy DB2 Entries panel 50
- copy utility jobs
 - generating 113
 - run after RELOAD or ALTER 113
- CREATE ALIAS 147
- CREATE apply job 113
- CREATE AUX TABLE 147
- CREATE DATABASE 147
- CREATE DISTINCT TYPE 147
- CREATE FUNCTION 147
- CREATE INDEX 147
- CREATE PROCEDURE 147
- CREATE SEQUENCE 147
- CREATE STOGROUP 147
- CREATE SYNONYM 147
- CREATE TABLE 147
- CREATE TABLESPACE 147
- CREATE TRIGGER 147, 158
- CREATE VIEW 147, 158
- creating an alternate input data set 211
- creating batch jobs 61
- creating version reports
 - from the DB2 catalog 153
- customization
 - associated list
 - adding DB2 entries 42
 - overview 234
 - associating DB2 entries 42
 - browsing parameters 50
 - changing display options 38
 - changing parameters 38
 - component 234
 - copying DB2 entries 50
 - Create a DB2 Entry panel 42
 - creating DB2 entries 42
 - customization jobs
 - deleting 53
 - displaying 53
 - generating 48
 - maintaining 53
 - regenerating 48
 - renaming 53
 - sort sequence 48
 - submitting 48, 53
 - customization library
 - deleting jobs 53
 - maintaining 53
 - overview 237
 - recustomizing 53
 - renaming jobs 53
 - customization library qualifier
 - specifying 34
 - Customized status 234
 - Customizer Workplace panel 48
 - customizing a new version of a product 38
 - customizing a product for the first time 38
 - customizing settings 34
 - data sets
 - customization library 237
 - data store 29, 237
 - DATASTOR 29
 - Discover EXEC library 237
 - metadata library 29, 237
 - customization (*continued*)
 - data store
 - overview 237
 - data store data set
 - specifying 34
 - DB2 data sharing members
 - adding 42
 - associating 42
 - copying 50
 - creating 42
 - DB2 entries 234
 - adding 42
 - associating 42
 - copying 50
 - creating 42
 - defining 48
 - deleting 52
 - generating jobs for 48
 - removing 52
 - selecting 48
 - specifying 48
 - unassociating 52
 - DB2 group attach field
 - specifying 34
 - DB2 group attach names
 - adding 42
 - associating 42
 - copying 50
 - creating 42
 - DB2 parameters
 - defining 46
 - editing 46
 - DB2 Parameters panel 46
 - DB2 subsystems
 - adding 42
 - associating 42
 - copying 50
 - creating 42
 - defining DB2 parameters 46
 - defining parameters 44, 48
 - defining product parameters 45
 - deleting DB2 entries 52
 - deleting jobs 40
 - Discover EXEC
 - customizing a new version of a product 38
 - overview 237
 - Discovered status 234
 - display options 38
 - displaying jobs 53
 - displaying panel text 38
 - editing parameters 38
 - editing product parameters 45
 - Errors in Customization status 234
 - finding trace data set 241
 - Finish Product Customization panel 48
 - first-time 39
 - first-time customization 38
 - generating jobs 48
 - hiding panel text 38
 - high-level qualifier 234
 - Incomplete status 234
 - job sort order 48
 - jobs
 - deleting 53
 - displaying 53
- customization (*continued*)
 - jobs (*continued*)
 - maintaining 53
 - renaming 53
 - sort order 48
 - submitting 48, 53
 - list of customization job members 233
 - LPARs 54
 - maintaining jobs 53
 - master list
 - adding DB2 entries 42
 - Associate DB2 Entry for Product panel 42
 - overview 234
 - maximizing information panels 38
 - metadata libraries
 - specifying 41
 - metadata library
 - maintenance, best practices 29
 - overview 237
 - specifying 34
 - modifying parameters 38
 - modifying settings 34
 - multiple instances 34
 - multiple-LPAR environment 54
 - Not Required status 234
 - options 38
 - panel display options 38
 - panels
 - Associate DB2 Entry for Product 42
 - Create a DB2 Entry 42
 - Customizer Workplace 48
 - DB2 Parameters 46
 - Finish Product Customization 48
 - Product Parameters 45
 - Specify the Metadata Library 41
 - parameters
 - browsing 50
 - defining 44, 48
 - viewing 50
 - preparing to use Tools Customizer 34
 - product 234
 - product parameters 233
 - changing 40
 - defining 45
 - editing 40, 45
 - modifying 40
 - Product Parameters panel 45
 - Ready to Customize status 234
 - recustomization 38, 40
 - recustomizing 40
 - recustomizing a product 38
 - removing DB2 entries 52
 - roadmaps 38
 - customizing for the first time 39
 - first-time customization 39
 - reustomizing 40
 - Specify the Metadata Library panel 41
 - specifying data sets 34
 - specifying metadata libraries 41
 - starting Tools Customizer 33
 - status types
 - Customized 234

- customization (*continued*)
 - status types (*continued*)
 - Discovered 234
 - Errors in Customization 234
 - Incomplete 234
 - Not Required 234
 - Ready to Customize 234
 - submitting jobs 48
 - terminology 234
 - trace data set 241
 - troubleshooting 241
 - finding trace data set 241
 - user job card settings
 - specifying 34
 - viewing parameters 50
- customization library
 - overview 237
- customization library qualifier
 - specifying 34
- customizing settings 34

D

- data set information field
 - batch job prefixes 113
 - PDS locations 113
- data sets
 - alternate values 211
 - ignoring 95
 - listing current values 211
 - modifying 95
 - naming conventions, using
 - ADB2UCUS 58
 - source version file data sets 79
 - specifying 90
- data store
 - maintenance, best practices 29
 - overview 237
- data store data set
 - specifying 34
- DB2 Administration Tool, accessing OCT
 - from 59
- DB2 databases, source input 69
- DB2 group attach field
 - specifying 34
- DB2 High Performance Unload 113
- DB2 Object Comparison Tool menu
 - (GOCMENU) panel 62
- DB2 objects, large number of 231
- DDL
 - DROP statements 158
- DDL file
 - source definitions 65
- deleting dialogs 62, 139
- dependencies
 - view dependent 158
 - view dropped 158
- diagnostic information
 - gathering 241
- dialogs
 - deleting 62, 139
 - managing 62, 139
 - renaming 62, 139
 - restoring 62, 139
 - saving 138
- Discover EXEC
 - overview 237

- display options 38
- displaying panel text 38
- Distributed DB2 Systems panel
 - (ADB2DDF) 140
- documentation
 - accessing 18
 - sending feedback 18
- documentation changes 1
- DROP ALIAS 147
- DROP apply job 113
- DROP DATABASE 147
- DROP DISTINCT TYPE 147
- DROP INDEX 147
- DROP SEQUENCE 147
- drop skip conversion 158
- DROP SPECIFIC FUNCTION 147
- DROP statements in the source
 - DDL 158
- DROP STORED PROCEDURE 147
- DROP SYNONYM 147
- DROP TABLE 147
- DROP TABLESPACE 147
- DROP TRIGGER 147
- DROP VIEW 147

E

- Edit Compare Masks (GOCEDIT)
 - panel 90
- Edit Compare Masks panel
 - (GOCEDIT) 90
- Eligible for auto-delete
 - stored compare results 107
- ERP (Enterprise Resource Planning) 231
- error messages
 - documentation for support 169
 - user action 169
- examples
 - ignore fields 194
 - translation masks 187
- exceptions to running compare jobs 140
- exclude from compare 140
- exclude specifications
 - creating 86
 - creating or managing in Change Management 84
 - Eligible for auto-delete
 - through Change Management 84
 - from stored compare results 86
- Exclude specifications
 - implicit drop 82
- excluding objects 82
- extracting source definitions 67

F

- features 3
- files
 - version 194
- first-time customization 39
- functions
 - overview 2
- functions, comparing 158

G

- general description 9
- generate apply job, alternate input 211
- Generate Compare Jobs panel
 - (GOC5) 105, 132, 140
- Generate Compare Jobs warning panel
 - (GOCGMPW) 113
- generate online 113
- generating a compare batch job 105
- generating a compare batch job
 - (multi-target import) 132
- generating a job online 113
- generating apply jobs
 - as work statement lists 113
 - generating a single apply job 113
 - generating multiple apply jobs 113
- generating templates 113
- generic ignore fields 196
- GOC1 64, 88
- GOC11 65
- GOC12 67
- GOC13 79
- GOC14 88
- GOC1C 77
- GOC1CA 77
- GOC1CD 77
- GOC1D 69
- GOC1DA 69
- GOC1DD 69
- GOC1S 72
- GOC1SA 72
- GOC1SD 72
- GOC1T 75
- GOC1TA 75
- GOC1TD 75
- GOC2MCMC 140
- GOC3 90
- GOC4 95
- GOC5 105, 132, 140
- GOC5AJ 113
- GOC5R0 113, 171
- GOC5RM 111
- GOC5WL 113
- GOC5I 95
- GOC5IF 95, 98, 99
- GOCEDIT 90
- GOCGMPW 113
- GOCMC 140
- GOCMC1 140
- GOCMCRL 140
- GOCMENU 62
- GOCMI 140
- GRANT collection privileges 147
- GRANT database privileges 147
- GRANT distinct type or JAR
 - privileges 147
- GRANT function or procedure
 - privileges 147
- GRANT package privileges 147
- GRANT plan privileges 147
- GRANT schema privileges 147
- GRANT sequence privileges 147
- GRANT table or view privileges 147
- GRANT use privileges 147

H

hiding panel text 38
High Performance Unload 113

I

ignore changes specifications
 Eligible for auto-delete
 through Change Management 104
 from stored compare results 101
 managing 102
 managing in Change
 Management 104
ignore fields
 examples 194
 explained 194
 generic 196
 selecting for SYSDATABASE 99
 selecting for SYSINDEXPART 98
 specifying 95, 168
 specifying in the Change Management
 database 95
 syntax 194
 system-generated 113
 user-specified 113
 user-specified ignore fields 113
 XMLMODIFIER 198
ignoring data sets 95
implicit drop 82
Implicit LOB and XML table
 support 131
include in compare 140
Insert Ignore panel (ADB2C22) 95
Insert Mask panel (ADB2C22) 90
interpret dialog 140
Interpret Dialog panel (GOCMI) 140
ISPF interface
 general description 5
 panel descriptions 61
ISPF table, editing 59

L

LABEL ON 147
legal notices
 cookie policy 301, 303
 notices 301
 programming interface
 information 301, 303
 trademarks 301, 303
listing current data set values 211
LOBs
 changing tables 165
 restriction 153
 unloading tables 165

M

M option 62, 139
main menu, configuration 59
Manage Dialogs MultiCompare panel
 (GOCMC) 140
Manage Dialogs MultiCompare panel
 (GOCMC) , line commands 140

Manage/Restore Dialogs (ADB2SDM)
 panel 139
managing changes
 comprehensive solutions 2
managing dialogs 62, 139
mask data sets 187
Mask Lines panel (ADB2CL2) 90
masks
 applying 187
 authorization ID 187
 authorization ID masks 187
 editing
 editing masks stored in the
 Change Management
 database 90
 examples 187
 hierarchy 192
 multi-target import 134
 name masks 187
 specifying 90
 specifying in the Change Management
 database 90
 syntax 187
 valid names 192
maximizing information on panels 38
MC option 62, 140
member prefix length
 building a batch job 113
 generating work statement list 136
metadata library
 maintenance, best practices 29
 overview 237
 specifying 34
modify source details 140
modify target details 140
modifying data sets 95
modifying settings 34
multi-target import
 masks 134
MultiCompare 62, 140
MultiCompare/Select (GOC2MCMC)
 panel 140
multiple compare objects 140

N

name masks 187
native SQL procedures
 comparing 158
 masking 158
notices 301, 303

O

Object authorizations 158
object count reports, changing reporting
 options 113
object types
 comparing 80
objects
 resetting 69, 75, 77
 selecting all
 from the Compare Add Databases
 panel 69
 from the Compare Add Schema
 panel 77

objects (*continued*)
 selecting all (*continued*)
 from the Compare Add Table
 Spaces panel 72
 from the Compare Add Tables
 panel 75
 source definitions 64
online job generation 113
online schema evolution
 limitations
 alter identity column
 attributes 158
 partition rotation 158
 partitioning key 158
optional jobs after reload
 REBIND 113
 REORG 113
 running REORG
 generating all REORGs to
 implement changes 113
 generating REORGs needed to
 remove pending conditions 113
optional jobs after RELOAD
 CHECK DATA 113
optional jobs after RELOAD or ALTER
 generating COPY utility jobs 113
 RUNSTATS 113
options 38
 apply change to work statement
 list 113
 generating apply jobs 113
 M 62, 139
 MC 62, 140
 member names 113
 R 62
 S 62, 138
 single compare jobs 113
 V 62
 W 61, 62
overview 1

P

panel display options 38
panels
 Batch Job Utility Parameters
 (ADB2UPA) 113
 Compare Add Databases
 (GOC1DA) 69
 Compare Add Databases
 (GOC1DD) 69
 Compare Add Schema (GOC1CA) 77
 Compare Add Schema (GOC1CD) 77
 Compare Add Table Spaces
 (GOC1SA) 72
 Compare Add Table Spaces
 (GOC1SD) 72
 Compare Add Tables (GOC1TA) 75
 Compare Add Tables (GOC1TD) 75
 Copy DB2 Entries 50
 DB2 Object Comparison Tool
 Menu 62
 described 61
 Distributed DB2 Systems
 (ADB2DDF) 140
 Edit Compare Masks (GOCEDIT) 90

panels (*continued*)

- Generate Compare Jobs (GOC5) 105, 132, 140
- Generate Compare Jobs warning (GOCGMPW) 113
- Insert Ignore (ADB2C22) 95
- Insert Mask (ADB2C22) 90
- Interpret Dialog (GOCMI) 140
- Manage Dialogs MultiCompare (GOCMC) 140
- Manage/Restore Dialogs (ADB2SDM) 139
- Mask Lines (ADB2CL2) 90
- MultiCompare/Select (GOC2MCMC) 140
- Register Options panel (ADB2CRO) 111
- Repeat Dialog/Change Location (GOCMRL) 140
- Save Dialog (ADB2SDS) 138
- Saved Dialogs (ADB2SD) 139
- Saved Dialogs (GOCMC1) 140
- Select Ignore Fields (GOCCIF) 95
 - selecting ignore fields for SYSDATABASE (GOCCIF) 99
 - selecting ignore fields for SYSINDEXPART (GOCCIF) 98
- Specify Compare Ignore Fields (GOC4) 95
- Specify Compare Masks (GOC3) 90
- Specify Compare Reporting Options (GOC5R0) 113, 171
- Specify Compare Source (GOC1) 64
- Specify Compare Target (GOC1) 88
- Specify Data Set Name for Apply Jobs (GOC5AJ) 113
- Specify DB2 Location (GOC14) 88
- Specify DB2 Source Catalog Extract (GOC12) 67
- Specify DDL File (GOC11) 65
- Specify Ignore Fields: Objects (GOCCI) 95
- Specify Job Parameters (ADB2W1R) 136
- Specify Register Mode panel (GOC5RM) 111
- Specify Source Compare Version File (GOC13) 79
- Specify Source DB2 Databases (GOC1D) 69
- Specify Source DB2 Schema (GOC1C) 77
- Specify Source DB2 Table Spaces (GOC1S) 72
- Specify Source DB2 Tables (GOC1T) 75
- Specify TEMPLATE UTILITY Usage (ADB25TU) 113
- Specify Work Statement List Data Set (GOC5WL) 113
- Utility Options (ADB2UOPS) 113

parallel unloads 113

partial database search names 69

partial schema search names 77

partial table search names 75

partition rotation 158

partitioned tables, comparing 158

partitioning key 158

prefix 138

preparing to use Tools Customizer 34

problems

- diagnostic information about 241

programming interface information 301, 303

programs

- ADB2GEN 153
- reverse engineering 153

R

R option 62

reader comment form 18

REBIND apply job 113

reference 233

Register Options panel (ADB2CRO) 111

registering changes 111

related tools 2

RELOAD apply job 113

removing objects 69, 75, 77

RENAME INDEX 147

RENAME TABLE 147

renamed object 158

renaming dialogs 62, 139

REORG

- generating to implement changes 113
- generating to remove pending conditions 113
- run after RELOAD or ALTER 113

REORG apply job 113

Repeat Dialog/Change Location panel (GOCMRL) 140

replicate dialog 140

replicate dialog multiple times 140

reports

- batch DB2 catalog extraction programs 153
- changing options 113
- creating versions from DB2 catalog 153
- format 171
- samples 171

resetting objects 69, 75, 77

resetting status 62

restoring dialogs 62, 139

restoring objects 9

results

- saved compare 107

retrieving dialogs 62

reverse engineering programs 153

roadmaps

- customizing for the first time 39
- first-time customization 39

RUN 140

RUNALL 140

running a work statement list 136

RUNSTATS, run after RELOAD or ALTER 113

S

S option 62, 138

sample reports 171

Save Dialog (ADB2SDS) panel 138

saved compare results 107

Saved Dialogs (ADB2SD) panel 139

Saved Dialogs panel (GOCMC1) 140

saving dialog collection names 138

saving dialogs 62, 138

scenarios 9

- comparing a DDL to a catalog 14
- comparing a development catalog to a production catalog 9
- modeling a new environment from an existing environment by copying objects 16
- undoing changes made in a catalog to catalog compare 13

schema, source input 77

screen readers and magnifiers 19

screens, described 61

Select Ignore Fields for SYSDATABASE panel (GOCCIF) 99

Select Ignore Fields for SYSINDEXPART panel (GOCCIF) 98

Select Ignore Fields panel (GOCCIF) 95

- selecting all objects 69, 72, 75, 77
- selecting ignore fields
 - for SYSDATABASE panel (GOCCIF) 99
 - for SYSINDEXPART panel (GOCCIF) 98
- generic 196
- XMLMODIFIER 198

selecting ignore fields for SYSCOLUMNS 100

service information 18

SET CURRENT PATH 147

SET CURRENT SQLID 147

single compare jobs 113

skip conversion 158

source 6

source DDL files 65

source details 140

- modifying 140
- viewing 140

source extractions 67

source input

- DB2 databases 69
- schema 77
- table spaces 72
- tables 75
- version file data sets 79

source objects 64

- specifying 167

Specify Compare Ignore Fields panel (GOC4) 95

Specify Compare Masks panel (GOC3) 90

Specify Compare Reporting Options (GOC5R0) 113, 171

Specify Compare Reporting Options panel (GOC5R0) 113

Specify Compare Source (GOC1) panel 64

Specify Compare Target panel (GOC1) 88

Specify Data Set Name for Apply Jobs (GOC5AJ) 113

Specify Data Set Name for Apply Jobs panel (GOC5AJ) 113

- Specify DB2 Source Catalog Extract panel (GOC12) 67
- Specify DDL File panel (GOC11) 65
- Specify Ignore Fields: Objects (GOCCI) panel 95
- Specify Job Parameters (ADB2W1R) 136
- Specify Register Mode panel (GOC5RM) 111
- Specify Source Compare Version File panel (GOC13) 79
- Specify Source DB2 Databases panel (GOC1D) 69
- Specify Source DB2 Schema panel (GOC1C) 77
- Specify Source DB2 Table Spaces panel (GOC1S) 72
- Specify Source DB2 Tables panel (GOC1T) 75
- Specify TEMPLATE UTILITY Usage panel (ADB25TU) 113
- Specify Work Statement List (ADB27WLD) 113
- Specify Work Statement List Data Set (ADB2WLDA) 113
- Specify Work Statement List Data Set panel (ADB2WLDA) 113
- Specify Work Statement List Data Set panel (GOC5WL) 113
- Specify Work Statement List panel (ADB27WLD) 113
- specifying
 - masks for a multi-target import 134
 - specifying compare ignore changes 168
 - specifying compare masks 167
 - specifying data sets 34
 - specifying ignore fields 168
 - specifying source objects 167
 - specifying target objects 167
- SQL statements
 - ALTER DATABASE 147
 - ALTER FUNCTION 147
 - ALTER INDEX 147
 - ALTER PROCEDURE 147
 - ALTER SEQUENCE 147
 - COMMENT ON 147
 - COMMIT 147
 - CREATE ALIAS 147
 - CREATE AUX TABLE 147
 - CREATE DATABASE 147
 - CREATE DISTINCT TYPE 147
 - CREATE FUNCTION 147
 - CREATE INDEX 147
 - CREATE PROCEDURE 147
 - CREATE SEQUENCE 147
 - CREATE STOGROUP 147
 - CREATE SYNONYM 147
 - CREATE TABLE 147
 - CREATE TABLESPACE 147
 - CREATE TRIGGER 147, 158
 - CREATE VIEW 147, 158
 - DROP ALIAS 147
 - DROP DATABASE 147
 - DROP DISTINCT TYPE 147
 - DROP INDEX 147
 - DROP SEQUENCE 147
 - DROP SPECIFIC FUNCTION 147
 - DROP STORED PROCEDURE 147
- SQL statements (*continued*)
 - DROP SYNONYM 147
 - DROP TABLE 147
 - DROP TABLESPACE 147
 - DROP TRIGGER 147
 - DROP VIEW 147
 - GRANT collection privileges 147
 - GRANT database privileges 147
 - GRANT distinct type or JAR privileges 147
 - GRANT function or procedure privileges 147
 - GRANT package privileges 147
 - GRANT plan privileges 147
 - GRANT schema privileges 147
 - GRANT sequence privileges 147
 - GRANT table or view privileges 147
 - GRANT use privileges 147
 - LABEL ON 147
 - RENAME INDEX 147
 - RENAME TABLE 147
 - SET CURRENT PATH 147
 - SET CURRENT SQLID 147
 - START apply job 113
 - status, resetting 62
 - STOP apply job 113
 - summary of changes 1
 - summary reports, changing reporting options 113
 - support
 - required information 241
 - support information 18
 - suppress DROP of objects 6
 - suppressing DROP, sequence 158
 - syntax
 - ignore fields 194
 - translation masks 187
 - SYSCOLUMNS, selecting ignore fields 100
 - SYSDATABASE, ignore fields 99
 - SYSINDEXPART, ignore fields 98
- T**
 - T10Ummn job 113
 - T20DROP job 113
 - T30CREAT job 113
 - T40STOP job 113
 - T50ALTER job 113
 - T60START job 113
 - T70Rnnnn job 113
 - T80Rnnnn job 113
 - T90RB job 113
 - table columns, comparing 158
 - table spaces, source input 72
 - tables
 - comparing partitioned 158
 - source input 75
 - target 6
 - target details 140
 - modifying 140
 - viewing 140
 - target objects
 - specifying 167
 - target objects, definitions 88
 - technotes 18
 - templates 113
- terminology
 - source 6
 - suppress DROP of objects 6
 - target 6
 - version file 6
- Tools Customizer
 - associated list
 - adding DB2 entries 42
 - overview 234
 - associating DB2 entries 42
 - browsing parameters 50
 - changing display options 38
 - component 234
 - Copy DB2 Entries panel 50
 - copying DB2 entries 50
 - Create a DB2 Entry panel 42
 - creating DB2 entries 42
 - customization jobs
 - deleting 53
 - displaying 53
 - generating 48
 - maintaining 53
 - renaming 53
 - sort sequence 48
 - submitting 48, 53
 - customization library
 - deleting jobs 53
 - maintaining 53
 - recustomizing 53
 - renaming jobs 53
 - customization library qualifier
 - specifying 34
 - Customized status 234
 - Customizer Workplace panel 48
 - customizing a new version of a product 38
 - customizing a product for the first time 38, 39
 - data sets
 - customization library 237
 - data store 29, 237
 - DATASTOR 29
 - Discover EXEC library 237
 - metadata library 29, 237
 - data store data set
 - specifying 34
 - DB2 data sharing members
 - adding 42
 - associating 42
 - copying 50
 - creating 42
 - DB2 entries 234
 - adding 42
 - associating 42
 - copying 50
 - creating 42
 - defining 48
 - deleting 52
 - generating jobs for 48
 - removing 52
 - selecting 48
 - specifying 48
 - unassociating 52
 - DB2 group attach field
 - specifying 34
 - DB2 group attach names
 - adding 42

Tools Customizer (*continued*)

- DB2 group attach names (*continued*)
 - associating 42
 - copying 50
 - creating 42
- DB2 parameters
 - defining 46
 - editing 46
- DB2 Parameters panel 46
- DB2 subsystems
 - adding 42
 - associating 42
 - copying 50
 - creating 42
- defining DB2 parameters 46
- defining parameters 44, 48
- defining product parameters 45
- deleting DB2 entries 52
- deleting jobs 40
- Discover EXEC
 - customizing a new version of a product 38
- Discovered status 234
- displaying jobs 53
- displaying panel text 38
- editing product parameters 45
- Errors in Customization status 234
- finding trace data set 241
- Finish Product Customization panel 48
- first-time customization 38, 39
- generating jobs 48
- hiding panel text 38
- high-level qualifier 234
- Incomplete status 234
- job sort order 48
- jobs
 - deleting 53
 - displaying 53
 - maintaining 53
 - renaming 53
 - submitting 53
- list of customization job members 233
- maintaining jobs 53
- master list
 - adding DB2 entries 42
 - Associate DB2 Entry for Product panel 42
 - overview 234
- maximizing information on panels 38
- metadata libraries 41
 - specifying 41
- metadata library
 - specifying 34
- multiple instances 34
- multiple-LPAR environment 54
- Not Required status 234
- panels
 - Associate DB2 Entry for Product 42
 - Copy DB2 Entries 50
 - Create a DB2 Entry 42
 - Customizer Workplace 48
 - DB2 Parameters 46
 - Finish Product Customization 48

Tools Customizer (*continued*)

- panels (*continued*)
 - Product Parameters 45
 - Specify the Metadata Library 41
- parameters
 - browsing 50
 - viewing 50
- preparing to use 34
- product 234
- product parameters 233
 - changing 40
 - editing 40
 - modifying 40
- Product Parameters panel 45
- Ready to Customize status 234
- recustomization 38
- recustomizing a product 38, 40
- removing DB2 entries 52
- roadmaps
 - recustomizing a product 40
- Specify the Metadata Library panel 41
- specifying metadata libraries 41
- starting 33
- status types
 - Customized 234
 - Discovered 234
 - Errors in Customization 234
 - Incomplete 234
 - Not Required 234
 - Ready to Customize 234
- submitting jobs 48
- terminology 234
- trace data set 241
- troubleshooting 241
- user job card settings
 - specifying 34
- viewing parameters 50
- trace data set
 - finding 241
- trademarks 301, 303
- translation masks
 - applying 187, 201
 - changing reporting options 113
 - examples 187
 - explained 187
 - hierarchy 192
 - syntax 187
 - valid names 192
- trigger creation, sequence 158
- triggers, comparing 158
- troubleshooting 169

U

- UNLOAD apply job 113
- unload methods
 - DB2 High Performance Unload 113
 - parallel 113
 - unload 113
- user job card settings
 - specifying 34
- user-defined template 227
- using alternate forms of syntax 6
- utilities, customizing options 113
- utility apply jobs 113
- Utility Options panel (ADB2UOPS) 113

V

- V option 62
- valid translation mask names 192
- version file 6
- version file data sets, source input 79
- version files 194
 - advantages 9
 - creating 8, 9
 - explained 8
- version files, comparing 158
- view source and target details 140
- viewing source and target details 140
- views, comparing 158

W

- W option 61, 62
- walking through steps 61, 62
- wildcards 69, 75, 77
- work statement lists
 - generating apply jobs 113
 - running 136
- worklist names 113

X

- XMLMODIFIER ignore fields 198



Product Number: 5655-DOC

Printed in USA

SC19-4139-05

