13.1

IBM Db2 Object Comparison Tool for z/OS User's Guide

SC28-2787-00

2023-11-16 edition

This edition applies to IBM® Db2® Object Comparison Tool for z/OS® 13.1 (product number 5655-CH1) and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

About this information

This information describes how to use IBM Db2 Object Comparison Tool for z/OS.

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 IBMDb2 objects by using Db2 Object Comparison Tool
- Generate reports and jobs by using Db2 Object Comparison Tool

Always check IBM Documentation (IBM Docs) for the most current version of this publication:

https://www.ibm.com/docs/en

Chapter 1. Overview of Db2 Object Comparison Tool

IBM Db2 Object Comparison Tool for z/OS (also referred to as Object Comparison Tool) compares existing Db2 for z/OS objects from different sources and reports the differences. Object Comparison Tool can subsequently synchronize these sources by making the relevant changes to the objects. Additionally, Object Comparison Tool is a required prerequisite for using the Change Management function of IBM Db2 Administration Tool for z/OS (Db2 Admin Tool).

Important: Db2 Object Comparison Tool 13.1 (5655-CH1) is available only as part of IBM Db2 Change Management Solution Pack for z/OS 1.2 (5655-CH1) and IBM Db2 Administration Solution Pack for z/OS 3.3 (5697-ASP). Object Comparison Tool is no longer available as a standalone product.

Specifically, Db2 Object Comparison Tool can help you with the following goals:

Keep your production system a mirror image of your test and development systems
New applications, changes to existing applications, or mistakes can cause Db2 objects in one system to have different attributes from objects in other systems. Object Comparison Tool can find differences between objects (and dependent objects) in a Db2 catalog on one system and a Db2 catalog on a different system. Object Comparison Tool can then generate batch jobs to synchronize the catalogs.

Compare objects with different names
Often, production objects and test objects use different naming conventions. You can account for these naming differences by using a feature in Object Comparison Tool called *masks*. With masks, object names can be translated before a comparison. Therefore, a test object can be matched to the corresponding production object for comparison. For example, if you want to compare all tables that begin with TEST to all tables that begin with PROD, you can define a mask that tells Object Comparison Tool to translate table names TEST* to PROD* for the comparison. (In this example, the asterisk is a wild card character.)

Ignore specific properties when comparing objects
You might not want your test objects to be exactly the same as your production objects. Object Comparison Tool can handle these intentional differences when comparing objects. To specify that the tool ignore certain attributes, such as the number of partitions in a table space or the storage group for a database, use the *ignore fields* feature.

Produce reports about the object comparison
Depending on the reporting options, Object Comparison Tool produces a variety of reports to show the differences between the objects.

Apply any changes to the target objects
Object Comparison Tool can generate jobs that apply any requested changes to the target objects. To request such jobs, use the *generate apply jobs* function. Alternatively, you can request that these changes be generated to a work statement list (WSL) that you can subsequently use to apply changes to the target objects. WSLs make it easy to propagate changes to remote sites.

Track changes
Changes can be imported into the Change Management Database to help you manage the process of recording and tracking the changes that you make to your objects.

Undo implemented changes
If you made changes and need to revert to the original state of the objects, Object Comparison Tool can revert those changes for you. The *undo capability* of the tool can restore application objects to a previous version.

Related information

IBM Db2 Administration Tool for z/OS

What's new in Db2 Object Comparison Tool 13.1

IBM Db2 Object Comparison Tool for z/OS (Object Comparison Tool) 13.1 introduces new features to support Db2 13 for z/OS as well as other usability features. Some of these enhancements were delivered on the General Availability (GA) date. Other enhancements were delivered later in the service stream, as part of new-function APARs.

Db2 13 function level support:

For information about any program temporary fixes (PTFs) that are required to support Db2 13 function levels, see Db2 13 function level support (IBM Db2 Administration Tool for z/OS 13.1.0).

GA enhancements in Db2 Object Comparison Tool 13.1

The following enhancements are available as of the General Availability (GA) date of Object Comparison Tool 13.1.

Online conversion of partition-by-growth (PBG) table spaces to partition-by-range (PBR) table spaces

Db2 13 introduces the capability to convert a table with growth-based partitions (in a PBG table space) to use range-based partitions (in a PBR table space) with an online change that has minimal impact to your applications. This online conversion is accomplished by using an ALTER TABLE statement with the new ALTER PARTITIONING TO PARTITION BY RANGE clause.

You can use Db2 Object Comparison Tool 13.1 to perform this online conversion. When APPLCOMPAT is set to V13R1M500 or higher and a target PBG table space needs to be changed to a PBR table space, Object Comparison Tool generates an ALTER statement when valid (according to any Db2 restrictions) and any necessary REORG statements to perform this conversion and thus minimize outages.

Related information:

Overview of what's new in Db2 13 (Db2 13 for z/OS documentation)
What's new in Db2 Admin Tool 13.1 (IBM Db2 Administration Tool for z/OS 13.1.0)

Support for package owner type

To increase flexibility for package ownership, Db2 13 allows you to specify whether the owner of a package is a role or authorization ID with the following new syntax:

- For the Db2 commands BIND and REBIND, Db2 13 introduces the new keyword OWNERTYPE for the OWNER bind option.
- For the SQL CREATE and ALTER statements for compiled SQL scalar functions and native SQL procedures, Db2 13 introduces the new keywords AS ROLE and AS OWNER in the PACKAGE OWNER clause.

Object Comparison Tool 13.1 can compare the owner and owner type for these procedures and functions and generate changes as needed. For example, if the owner and owner type differ between the source object and the target object, the compare report contains a message similar to the following message:

```
Options
  (A)Field PACKAGE OWNER changed from 'RL174061 AS ROLE' to 'TS5465 AS USER'
Native SQL Procedure options will be altered
```

Related information:

Overview of what's new in Db2 13 (Db2 13 for z/OS documentation)
What's new in Db2 Admin Tool 13.1 (IBM Db2 Administration Tool for z/OS 13.1.0)

Support for long column names

Db2 13 introduced support for long column names (up to 128 bytes) when the TABLE_COL_NAME_EXPANSION subsystem parameter setting is ON. Previously, the limit was 30 bytes. Object Comparison Tool 13.1 can manage these longer column names. For example, you can compare objects with long column names and generate changes as needed.

Related information:

Overview of what's new in Db2 13 (Db2 13 for z/OS documentation)
Column names longer than 30 bytes (Db2 13 for z/OS documentation)
What's new in Db2 Admin Tool 13.1 (IBM Db2 Administration Tool for z/OS 13.1.0)

Related reference

"New-function APARs in Db2 Object Comparison Tool 13.1" on page 3
After GA, enhancements continue to be delivered later in the service stream, as part of new-function APARs.

Related information

Db2 13 function level support (IBM Db2 Administration Tool for z/OS 13.1.0)

New-function APARs in Db2 Object Comparison Tool 13.1

After GA, enhancements continue to be delivered later in the service stream, as part of new-function APARs.

The following table summarizes the APARs that introduce new function for Object Comparison Tool 13.1. It does not include problem fixes or other maintenance APARs.

Description	APAR	Date
"Ability to compare APPLCOMPAT for expression-based indexes" on page 4	PH57692 PH57696 PH57698	2023-11
"Ability to use a SELECT statement to specify source and target objects" on page 4	PH57198 PH57326	2023-10
"New CM batch options to include foreign keys changes when comparing objects" on page 4	PH55583	2023-10
"Summary report can include original names of added objects" on page 5	PH56749	2023-09
"Support for regenerating views" on page 5	PH37650 PH55431	2023-07
"CM batch support for comparing DDL to DDL" on page 5	PH54480	2023-05
"Masking support for removing a key label" on page 6	PH54152	2023-05
"Improvements when transporting work statement lists to other systems" on page 6	PH53482	2023-03
"Eliminate unnecessary changes when comparisons involve objects created prior to Db2 12" on page 6	PH49601	2022-11
"Ability to specify REBIND options when altering objects" on page 7	PH50333	2022-11
"REORG SHRLEVEL default change to avoid pending changes" on page 8	PH49639	2022-09

Description	APAR	Date
"Improvements to inserting and adding partitions" on page 8	PH48016	2022-08

Ability to compare APPLCOMPAT for expression-based indexes

PH57692 (Db2 Admin Tool), PH57696 (Object Comparison Tool), PH57698 - November, 2023

When you compare expression-based indexes, the APPLCOMPAT values are compared. If you do not want to compare these values, you can now ignore them by using the new ignore SYSENVIRONMENT APPLCOMPAT.

Db2 Admin Tool also now provides the ability to manage and change APPLCOMPAT values for expression-based indexes. For more information about these enhancements, see 2023 new-function APARs for Db2 Admin Tool 13.1 (IBM Db2 Administration Tool for z/OS 13.1.0).

Related information:

"4. Specifying ignores" on page 73
PH57692
PH57696

Ability to use a SELECT statement to specify source and target objects

PH57198, PH57326 - October, 2023

In Db2 Object Comparison Tool, you specify the objects to be compared by selecting either DDL, objects from the Db2 catalog, or a compare version file. With this enhancement, you now have another option: you can identify the source and target objects to be compared by using an SQL SELECT statement against the Db2 catalog. Object Comparison Tool uses all of the objects that are returned by the query for the source or target definition.

Using a SELECT statement to identify the source and target objects can be more efficient than individually specifying objects from the Db2 catalog. Additionally, when using a SELECT statement, you can use clauses to easily filter the object list.

To specify a SELECT statement, you must first select the Db2 catalog as the source of the object definitions [option 2 on the **Specify Compare Source (GOC1)** panel or the **Specify Compare Target (GOC1)** panel]. Then you can select the new option: **5 - Source is the result of an SQL SELECT statement**. The SQL statement that you specify must return certain columns and can optionally return other columns. For detailed instructions and requirements, see "Specifying a SELECT statement for the source or target definition" on page 56.

You can also specify a SELECT statement when using Change Management (CM) batch to run a comparison. In this case, set the SOURCE_TYPE parameter, TARGET_TYPE parameter, or both to USER and use a quick scope to specify the SELECT statement. For detailed instructions and examples, see Chapter 12, "Running Compare by using a Change Management batch job," on page 155.

Related information:

PH57198
PH57326

New CM batch options to include foreign keys changes when comparing objects

PH55583 - October, 2023

When running the CM batch interface JCL procedure (GOCCM) to compare objects, you can now specify whether foreign key changes should be included in the generated DDL. To do so, use the following new CM batch parameters:

- SOURCE_GEN_FOREIGN_KEYS
- TARGET_GEN_FOREIGN_KEYS

This functionality is similar to the GENRELS parameter in the JCL that is generated from the **Generate Compare Jobs (GOC5)** panel.

Related information:

CM batch parameter definitions (IBM Db2 Administration Tool for z/OS 13.1.0)
PH55583

Summary report can include original names of added objects

PH56749 - September, 2023

When you compare objects with Object Comparison Tool, the resulting changes might include adding objects to the target. In this case, those new object names might be masked and therefore different than the original object names in the source. To help you determine which source object was added, you can now request that Object Comparison Tool report the original object name in addition to the new name when running change management (CM) batch. To do so, set the new CM batch parameter REPORT_ORIGINAL_NAMES_ADDED_OBJECTS to YES. When this parameter is set to YES, the summary report includes the original name (under `Source Object`) and the new masked name (under `Target Object`), as shown in the following example:

```
COMPARISON SUMMARY REPORT
=========================

Obtyp Source Object                   Target Object                   Result
Object type
----- ------------------------------- ------------------------------- --------------------
---------------
X     MPX8130.XDEPTNEW                MPT8130.XDEPTNEW                Added                Index
```

Related information:

CM batch parameter definitions (IBM Db2 Administration Tool for z/OS 13.1.0)
PH56749

Support for regenerating views

PH37650, PH55431 - July, 2023

Object Compare now supports regenerating views if needed when the source and target views have different APPLCOMPAT values. Specifically, if the source and target views are the same, but the APPLCOMPAT values are different, and the source view APPLCOMPAT value is lower than current APPLCOMPAT value, Object Compare generates the following statement:

```
ALTER VIEW view_name REGENERATE USING APPLICATION COMPATIBILITY source_applcompat
```

Related information:

PH37650
PH55431

CM batch support for comparing DDL to DDL

PH54480 - May, 2023

You can now compare DDL to DDL by using Change Management (CM) batch. Previously, this type of comparison was allowed only by using the Object Comparison Tool panels. To do this comparison, use the new DDL value for the TARGET_TYPE CM batch parameter. For details, see 2023 new-function APARs for Db2 Admin Tool 13.1 (IBM Db2 Administration Tool for z/OS 13.1.0).

Related information:

PH54480

Masking support for removing a key label

PH54152 - May, 2023

When comparing objects, you can overwrite a key label value for a storage group or table by using the existing masks SGKEYLABL and TBKEYLABL, respectively. This APAR adds support to these masks for removing a key label. You can now specify the value NOKEYLABEL (or NO) for these masks to remove a key label. For example:

```
TBKEYLABL:TBCRE.MYTB,NOKEYLABEL**
TBKEYLABL:TBCRE.MYTB,NO**
```

Related information:

"Mask data set" on page 68
PH54152

Improvements when transporting work statement lists to other systems

PH53235 (Db2 Admin Tool), PH53482 (Object Comparison Tool) - March, 2023

When you compare objects by using Db2 Object Comparison Tool, you can store the resulting changes in a work statement list (WSL). If those changes require an unload operation, Object Comparison Tool generates an IFF file. If you then transport the WSL to another system, the IFF file must also be transported independently. To simplify this process of transporting WSLs, a new option is now available to embed the IFF file directly in the WSL. When you specify this option, all of the information in the IFF file is embedded in the WSL in an encoded format. You can then easily transport the WSL to another system without needing a separate IFF file.

This new **Embed IFF into WSL** option is available on the **Generate Compare Jobs (GOC5)** panel when comparing objects. The default value for this new option is NO.

This option is also available on several Db2 Admin Tool panels. See 2023 new-function APARs for Db2 Admin Tool 13.1 (IBM Db2 Administration Tool for z/OS 13.1.0).

Related information:

"Compare job options" on page 80
PH53235
PH53482

Eliminate unnecessary changes when comparisons involve objects created prior to Db2 12

PH49601 - November, 2022

If a table space was created prior to Db2 12, certain table space attributes that are new in version 12 might still be set to NULL in the Db2 catalog for that table space. If you use Db2 Object Comparison Tool to compare one of these table spaces in the catalog (the target) with DDL that was created in Db2 12 or later (the source), and the first partition values on the target are the same as the source table-space-level attributes, no ALTER statements should be generated for these attributes.

This APAR ensures that Object Comparison Tool does not generate unnecessary changes for the following attributes when comparing table spaces and the target is a table space in the catalog that was created prior to Db2 12:

- PQTY
- SECQTYI
- STORTYPE
- STORNAME
- VCATNAME

- PCTFREE
- PCTFREE_UPD
- TRACKMOD
- COMPRESS
- FREEPAGE
- GBPCACHE

A similar situation exists for indexes that were created prior to Db2 12. This APAR also ensures that Object Comparison Tool does not generate unnecessary changes for the following index attributes when comparing indexes and the target is an index in the catalog that was created prior to Db2 12:

- PQTY
- SECQTYI
- STORTYPE
- STORNAME
- VCATNAME
- FREEPAGE
- PCTFREE
- GBPCCAHE

Related information:

PH49601

Ability to specify REBIND options when altering objects

PH50333 - November, 2022

When a comparison results in a change that requires an object to be altered, Db2 Object Comparison Tool lets you choose whether to rebind any dependent packages. Prior to this APAR, these packages were rebound with their existing BIND options (the options that were used during the previous bind or rebind operation). With this APAR, you can now specify different BIND options. For example, you can specify APREUSE(ERROR) to help retain existing access paths.

To specify REBIND options, use the new **REBIND options** field on the **Generate Compare Jobs (GOC5)** panel:

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

 Specify the following for DB2 Object Comparison Tool:
 ....

     For ROWID . . . . . . . . NO       (Yes/No)
     For ROW CHANGE TIMESTAMP. NO       (Yes/No)
   Retain START and RESTART values:
     For sequence object . . .          (Yes/No)
   IDENTITY START value  . . . ORIGINAL (Original, Computed)
   Mask ignored fields . . . . NO       (Yes/No)

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

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

When you specify Yes in this new field (and either M or A in the **Run REBIND** field), the **REBIND options (ADBPREBO)** panel is displayed where you can specify the following options:

```
ADBPREBO  ----------------------- REBIND options -----------------------
Command ===>


Specify additional REBIND parameters to generate rebinds
for dependent packages.

  APREUSE . . . .                (None, Warn, Error)
  EXPLAIN . . . .                (Yes, No, Only)
  OWNER . . . . .              > (Owner of package)
    OWNERTYPE . .                (Role, User)

  Additional options:
_____________________________________________________


_____________________________________________________________________________


_____________________________________________________________________________


_____________________________________________________________________________

  _____________________________________________________________________________
```

Any BIND options that you specify in the **Additional options:** field are added as is; they are not validated.

Related information:

“5. Generating a compare batch job” on page 78
PH50333

REORG SHRLEVEL default change to avoid pending changes

PH49639 – September, 2022

When a comparison results in a change that requires a REORG utility operation and no value is specified for the REORG SHRLEVEL option, Object Comparison Tool generates a REORG statement with a default value for SHRLEVEL. With this APAR applied, SHRLEVEL NONE will no longer be generated for this situation, because it prevents pending definition changes from being materialized and can leave objects in a pending state. Instead, to ensure that any pending changes are materialized successfully, either SHRLEVEL CHANGE or SHRLEVEL REFERENCE will be used; Object Comparison Tool determines the best value (CHANGE or REFERENCE) depending on the circumstance.

As usual, you can override this behavior by specifying a value for SHRLEVEL and setting **Use customized util opts** to YES on the **Generate Compare Jobs (GOC5)** panel. If you specify SHRLEVEL NONE and a pending change exists, a warning is issued.

This change also applies to Change Management and the ALT command in Db2 Admin Tool. For more details, see New-function APARs for Db2 Admin Tool 13.1 (IBM Db2 Administration Tool for z/OS 13.1.0).

Related information:

PH49639

Improvements to inserting and adding partitions

PH48016 – August, 2022

Db2 Object Comparison Tool is enhanced to improve how partitions are added and inserted. In some cases, these changes reduce unnecessary and potentially costly REORG utility operations.

Prior to this APAR, when comparing partition-by-range (PBR) table spaces where the source has more partitions than the target, Object Comparison Tool generated the following statements for each partition to be added:

1. ALTER TABLE statement with the ADD PARTITION (MAXVALUE) clause.
2. REORG utility statement.
3. ALTER TABLE statement with the ALTER PARTITION clause to alter the added partition with the new limit key.

With this APAR applied, a single ADD PARTITION is generated, which also reduces the REORG statements that are generated in some cases.

Related information:

PH48016

The comparison process

Db2 Object Comparison Tool compares Db2 objects (and dependent objects) and reports the differences. As part of this process, Object Comparison Tool can optionally generate jobs to modify the objects to eliminate these differences.

The two sets of Db2 objects that are compared are called *source* and *target* objects. A *source object* is an object as you want it defined. A *target object* is an object that you want to match the source object.

When you select the source and target objects to compare, Object Comparison Tool extracts definitions of the objects and places them in a sequential data set called a *version file*. A *version file* is an internal representation of a set of objects and represents a snapshot at a particular point in time. Two separate version files are created, one for the source object and one for the target object. These version files are created before the objects are compared.

You can specify any of the following sources that you want Object Comparison Tool to use for the object definition:

DDL file
A file that contains data definition language (DDL), such as a SPUFI file. When the source is DDL, Object Comparison Tool processes everything in the DDL file. Objects are not selected based on type or name. If you are comparing DDL and your DDL only defines a table, only that table is used.

Db2 catalog
An extract of information from the Db2 catalog for one or more databases, table spaces, or tables and all the dependent objects. When the definition source is a Db2 catalog, Object Comparison Tool includes all dependent objects, such as views and indexes, in the comparison. These dependent objects are included regardless of whether you specify objects at the database level, the table space level, or the table level.

Version file
A version file that was created during a previous comparison. If a version file is used as the source of the comparison, a new version file is not created.

Using Object Comparison Tool, you can do any of the following comparisons:

Definition source for the source object	Definition source for the target object
Db2 catalog	Db2 catalog
DDL file	DDL file
Version file	Version file
DDL file	Db2 catalog
DDL file	Db2 catalog with objects that are automatically selected based on the source specification

After the source and target version files are created, Object Comparison Tool compares them and creates a difference file or *changes file*. Object Comparison Tool then generates reports that show the differences between the objects and, if requested, DDL to apply any changes to the target object. After reviewing the

report, you can direct the generated DDL for the target object to apply jobs. Apply jobs can be stored in a work statement list (WSL) or a partitioned data set (PDS) and then propagated to several remote sites. This process allows for changes in a test environment to be easily migrated to the development or production environment.

As part of the comparison process, you can use masking and ignore fields to account for intentional differences between the objects, so that only the actual differences are reported. *Masking* handles different naming conventions between the objects that you are comparing. For example, the same object might have an owner name of TEST*xxx* on the test system and an owner name of PROD*xxx* on the production system. *Ignore fields* handle attribute differences between the objects that you are comparing. For example, primary and secondary quantities usually differ between test and production systems.

Related concepts

“Components of the comparison process ” on page 10
Db2 Object Comparison Tool compares objects by reading the Db2 catalog or DDL files. Object Comparison Tool produces comparison reports and then optionally generates either JCL jobs or work statement list (WSL) tasks with changes for the target objects.

Components of the comparison process

Db2 Object Comparison Tool compares objects by reading the Db2 catalog or DDL files. Object Comparison Tool produces comparison reports and then optionally generates either JCL jobs or work statement list (WSL) tasks with changes for the target objects.

The following figure shows the detailed flow of processes in Db2 Object Comparison Tool:

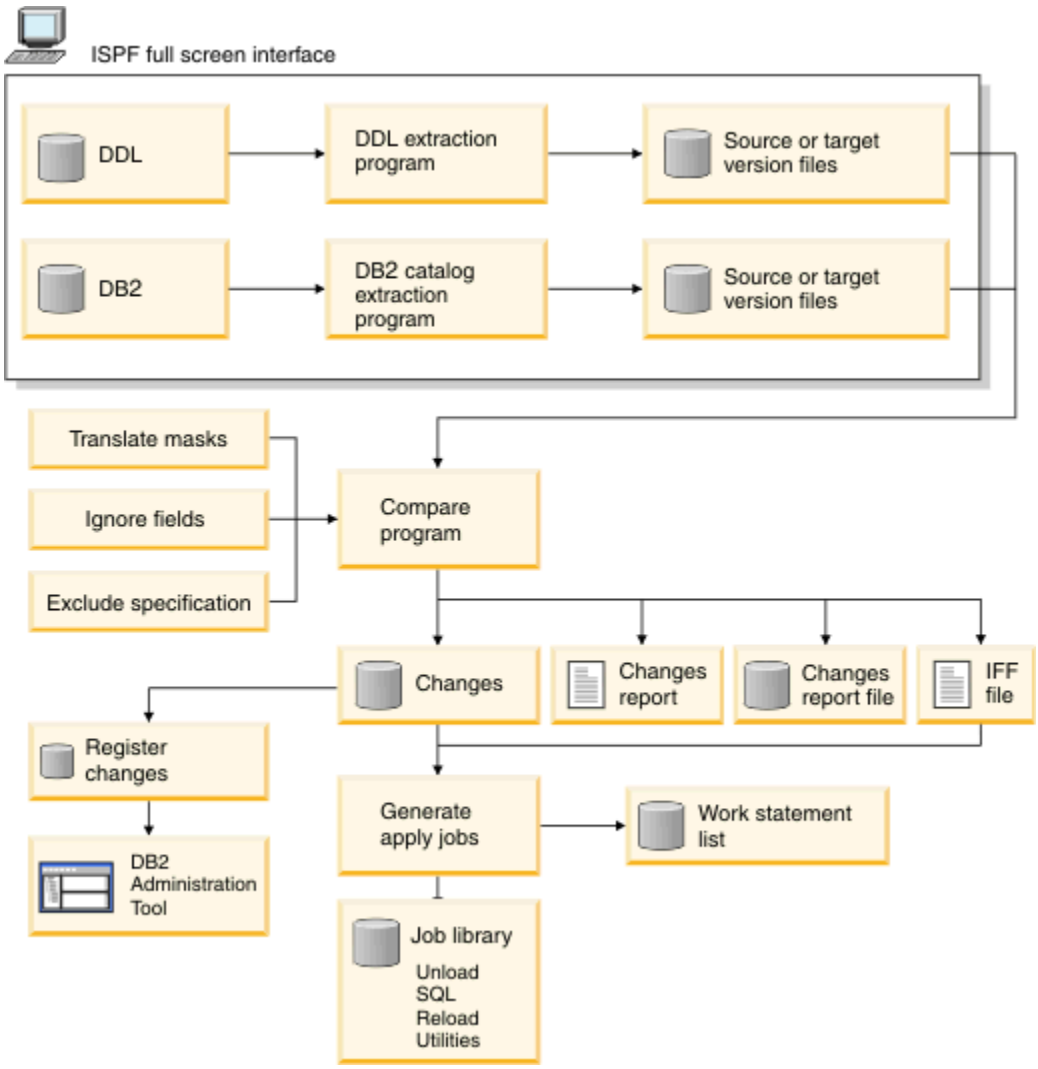

Figure 1. Db2 Object Comparison Tool processes and components

This figure includes the following processes and components:

DDL extraction program
This program reads object definitions from DDL files into a version file.

Db2 catalog extraction program
This program reads object definitions from the Db2 catalog into a version file.

Compare program
This program compares two version files, produces a report to describe any differences, and generates the information that is needed to apply changes to the target object. This program accounts for any specified masks, ignore fields, or exclude specifications when doing the comparison.

Program to register the changes
This program registers the changes in the Change Management (CM) database in Db2 Administration Tool, where you can then analyze and run the job.

Function to generate apply jobs
This function performs one of the following operations:

- Creates the UNLOAD, DROP, CREATE, ALTER, and LOAD jobs that are necessary to apply the changes to the target object.
- Creates WSL tasks to apply the necessary changes to the target object.

Related concepts

"The comparison process" on page 9

Db2 Object Comparison Tool compares Db2 objects (and dependent objects) and reports the differences. As part of this process, Object Comparison Tool can optionally generate jobs to modify the objects to eliminate these differences.

Terminology in Db2 Object Comparison Tool

Db2 Object Comparison Tool uses several terms that are unique to the product.

Alternate form of syntax
Another acceptable syntax for a statement.

Certain functions in Object Comparison Tool and Db2 Administration Tool (Db2 Admin Tool) support or produce statements that are used by Db2 for z/OS or by these two products. IBM might provide an alternate statement or alternate form for clauses in statements. IBM might identify one as the preferred syntax while still supporting the alternate form.

Object Comparison Tool and Db2 Admin Tool 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. When necessary to produce an accepted statement, the products convert to the newer syntax. However, the products might retain older syntax even if Db2 considers the newer syntax the preferred syntax. This situation might be the case even if no possible use of the older syntax is needed. The use of older syntax might persist until IBM no longer supports it.

Changes file
The file that Object Comparison Tool creates when the source and target objects are compared. This file is used by Object Comparison Tool to generate a report of the differences between the objects. This file is also used by the generate apply jobs function.

The changes file contains the following items:

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

The name of a typical changes file might be NBRON.PQ76055N.CHANGES.

Exclude
A specified object or authorization to exclude from input to the compare process.

Exclude Specification
A specification that lists objects that you want to exclude from the compare process.

Interchange File Format (IFF) file
A file that is produced by the compare program. This file and the changes file are used by Object Comparison Tool to generate the apply jobs.

Ignore change
A specified change to an object that you want to ignore.

Ignore change specification
A specification that identifies changes that you want ignored during the comparison process. You can select the changes that you want ignored from a saved comparison report. Object changes that you specify as ignored are reported, but no SQL statements are generated for the changes.

Ignore fields
Fields that Object Comparison Tool ignores when comparing Db2 catalog records.

Source
The structure of the objects as you want them to look. For example, the source can be the structure of objects in a development environment. The source can be from DDL, a version file, or the Db2 catalog.

Suppress DROP of objects
An option that prevents dropping objects that exist in the target but not in the source.

By default, Object Comparison Tool drops 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, Object Comparison Tool drops object B. This behavior is the default.

To change this default behavior, set the **Suppress DROP of objects** option to `Yes`. Generally, you should set this option to `Yes` if your source is a subset of the target and you want to avoid possible dropped objects. For example, if you specify DDL as the source and a database in the Db2 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 are dropped unless you specify `Suppress DROP of objects =Yes`.

Target
The destination for the changes. For example, the target can 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 definition can be an explicit specification of DDL, a version file, or the Db2 catalog, or an implicit selection of objects based on the source.

In the 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
A functionality that allows a match to be found when the source and target objects use different naming conventions. Before 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. Object Comparison Tool creates a version file for each source and target and then uses those files to perform a comparison.

A version file is a variable-length data set that contains all the information that was extracted about the Db2 objects. 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 also keep multiple records for the same object together.

Version files can be saved for subsequent comparison operations. You can also use them to restore application objects to a previous version (undo) or compare a new version with several production versions (clones) of the objects.

Product documentation and updates

The documentation for Db2 Object Comparison Tool is regularly updated with information about new features and any corrections.

The Object Comparison Tool documentation is available in the following two formats:

Topics in IBM Documentation
Underneath the title of each topic, you can see the date it was last updated.

You can find IBM Db2 Object Comparison Tool for z/OS in IBM Documentation at https://www.ibm.com/docs/en/db2objectcompare

Tip: When searching IBM Documentation, use quotation marks to ensure exact matches only. For example, the search term `"ADB226E"` returns only those topics that contain ADB226E. If you do not use quotation marks, close or partial matches might be returned. For example, a search on `ADB001E` might return ADB901E. However, a search on `"ADB001E"` returns no results.

PDF format
The PDF is titled "IBM Db2 Object Comparison Tool for z/OS User's Guide." The date when the PDF was created is listed at the bottom of page 2, near the copyright information.

The latest copy of the PDF is always posted at https://www.ibm.com/docs/en/SSAUVH_13.1.0/pdf/gocug131.pdf.

Both of these formats contain the same information and are updated at the same time.

Revision marks for changed content

Revisions for the following types of content changes are marked like this sentence, with black bars in the left margin:

- Technical revisions for changed externals that are introduced by the new release or by maintenance after the general availability of this release.
- Technical clarifications in response to customer and internal feedback.

Editorial and organizational changes that do not affect the technical meaning of the content are generally not marked.

How to send your comments

Your feedback is important in helping to provide accurate and high-quality information. If you have any comments about this information or any other IBM product documentation, send your comments to ibmdocs@us.ibm.com.

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 documentation for the specific assistive technology for information about 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), the default settings for the PF keys, and how to modify their functions.

Chapter 2. Customization

When you customize Db2 Admin Tool, you can also enable 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.

To customize Object Comparison Tool, use IBM Tools Customizer for z/OS 1.1 (5655-TC1), also known as TCz. Formerly a component of IBM Tools Base, TCz is a standard tool for customizing IBM tools that run on z/OS. It provides a single, common, and consistent ISPF interface for post-installation customization of these tools.

The instructions in this section are specific to Object Comparison Tool. For detailed information about how to use TCz, see IBM Tools Customizer for z/OS 1.1.0.

Migration to a new Db2 version, mode, or function level: When you migrate to a new Db2 version, mode, or function level, you do not need to recustomize Object Comparison Tool. Because the product relies on Db2 Admin Tool to access Db2, you need only recustomize Db2 Admin Tool.

Customization checklist for Db2 Object Comparison Tool

The following checklist describes each significant customization step. Use this checklist to guide you through the entire customization process for Object Comparison Tool.

Tips:

- Print this checklist and record your status during the customization process.
- If you are not familiar with Tools Customizer (TCz) and the customization process, consider reviewing the following terminology and other basic TCz information before you begin: Tools Customizer terminology and data sets (IBM Tools Customizer for z/OS 1.1)

Task	Link to detailed instructions	Status
Verify software requirements		
Verify that your environment meets the minimum software requirements.	“Software requirements for Object Comparison Tool” on page 16	
Verify SMP/E installation		
Verify that Object Comparison Tool is installed. SMP/E installation instructions are in the program directory. To verify that the installation completed correctly, specify the following command on any Db2 Admin Tool panel: `PANEL GOCMENU` The **DB2 Object Comparison Tool Menu (GOCMENU)** panel should be displayed. If this panel is not displayed, the installation was not successful and you must reinstall Object Comparison Tool.	Program Directory for Db2 Object Comparison Tool 13.1 (GI13-4643)	
Verify that TCz is installed. SMP/E installation instructions are in the program directory.	Program Directory for IBM Tools Customizer for z/OS 1.1 (GI13-4653)	
Gather data set names		

Task	Link to detailed instructions	Status
Record the data set names that you will need during the customization process.	"Data sets used by Tools Customizer" on page 17	
Optional: Determine LPAR strategy		
If you have a multiple-LPAR environment, determine your customization strategy.	"Using Tools Customizer in a multiple-LPAR environment" on page 21	
Customize Db2 Object Comparison Tool		
Complete the steps in the appropriate customization roadmap based on the type of customization that you are performing.	"Roadmap: Customizing Db2 Object Comparison Tool for the first time" on page 17 "Roadmap: Recustomizing Db2 Object Comparison Tool" on page 20	
Allocate libraries		
Before you can use Db2 Object Comparison Tool, you must allocate the libraries to your ISPF session.	"Allocating libraries for Db2 Object Comparison Tool" on page 26	
Optional: Customize JCL		
Customize the JCL that Object Comparison Tool uses to adhere to your installation standards.	"Customizing the JCL that Object Comparison Tool uses" on page 26	
Optional: Customize data set names		
Align the Db2 Admin Tool data set names with your local data set naming conventions.	"Customizing data set names" on page 27	
Optional: Make Object Comparison Tool available from Db2 Administration Tool		
When you customize Db2 Admin Tool, you can make Db2 Object Comparison Tool available from the main menu.	Making DB2I and IBM Db2 Object Comparison Tool for z/OS available from the Db2 Administration Tool main menu (IBM Db2 Administration Tool for z/OS 13.1.0)	
Optional: Enable product discovery		
This step is strongly recommended if Db2 Administration Foundation is also installed.	"Enabling product discovery for Object Comparison Tool" on page 29	

Preparing to customize Db2 Object Comparison Tool

Before you use TCz to customize Object Comparison Tool, review the software requirements and gather the information that you will need.

Software requirements for Object Comparison Tool

Prior to beginning the customization process for Object Comparison Tool, ensure that your environment meets all software requirements.

Object Comparison Tool 13.1 requires the following software:

- One of the following supported versions of Db2 for z/OS:
  - Db2 13 (5698-DB2)
  - DB2® Value Unit Edition 13.1 (5698-DBV)
  - 5650-DB2

  - Db2 Value Unit Edition 12.1
- The requisite release of z/OS for the Db2 subsystems that you will be using with Object Comparison Tool
- IBM Db2 Administration Tool for z/OS 13.1 (5698-AT3).

Memory recommendations: Because Db2 Object Comparison Tool keeps information in memory for efficiency, use a minimum region of 256 MB of memory for both batch and TSO. Ideally, if allowed by your installation policy, set REGION=0M for batch jobs to allow for maximum below-the-bar storage and avoid reruns.

When 1000 or more objects are processed, additional region is recommended. More memory is also necessary if you suppress object dropping when generating the job, because object attributes are kept resident to process this option. If you are processing more than 10,000 objects, use a starting region of 256 MB. If LE storage failures occur, increase region parameters before assuming that a problem exists. Increase memory in 32 MB increments.

In all cases, ensure that the requested region size is not limited to a lower amount by the IEFUSI installation exit.

Data sets used by Tools Customizer

Tools Customizer (TCz) uses the following data sets during the customization process:

Data set name	Description
SCCQEXEC	EXEC library for TCz
SCCQDENU	Metadata library for TCz
SCCQLOAD	Executable load module library for TCz
SCCQMENU	ISPF messages for TCz
SCCQPENU	ISPF panels for TCz
SCCQSAMP	Sample members for TCz
SCCQTENU	Table library for TCz

Customizing Db2 Object Comparison Tool

After Db2 Object Comparison Tool is installed, you can customize the configuration by running IBM Tools Customizer for z/OS (TCz).

For an overview of the entire process, see “Customization checklist for Db2 Object Comparison Tool” on page 15.

Roadmap: Customizing Db2 Object Comparison Tool for the first time

When you install Db2 Object Comparison Tool for the first time, you must customize the configuration by using IBM Tools Customizer for z/OS (TCz).

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

Tip:

- For multiple-LPAR environments, determine your customization strategy first: “Using Tools Customizer in a multiple-LPAR environment” on page 21.
- For guidance on any input fields in TCz, position your cursor on the input field and press F1 (Help).

Table 1. Steps for customizing Object Comparison Tool for the first time

Step	Description	Instructions

Start Tools Customizer.	1. Edit the CCQTCZ member in the *hlq*.TCZ110.SCCQEXEC data set.
	2. Locate TCZHLQ="<TCz HLQ>".
	3. Change "<TCz HLQ>" to the high-level qualifier of your TCz EXEC data set, as shown in the following example:
	`TCZHLQ="hlq.TCZ110"`
	4. Save your changes.
	5. On the ISPF Command shell panel, issue the following command:
	`EX 'hlq.TCZ110.SCCQEXEC(CCQTCZ)'`
		• Starting Tools Customizer (IBM Tools Customizer for z/OS 1.1)
		• "Data sets used by Tools Customizer" on page 17

Modify Tools Customizer settings.	1. On the **IBM Tools Customizer for z/OS (CCQPHME)** panel, specify option 0 (**User settings for Tools Customizer**). 2. Specify values for the following required sections: • Customization library qualifier • Use Db2 group attach name • Metadata library • Discover output data set • Data store data set • User job card settings 3. Save your changes, and press Enter.	Modifying Tools Customizer user settings (IBM Tools Customizer for z/OS 1.1)
Specify the Object Comparison Tool metadata library.	1. On the **IBM Tools Customizer for z/OS (CCQPHME)** panel, specify option 1 (**Customize a product**). 2. On the **Specify the Product or Pack Metadata Library (CCQPHLQ)** panel, enter the following value in the **Product or pack metadata library** field, and press Enter: `DMTOOL.SGOCDENU`	Specifying the metadata library for the product or pack to customize (IBM Tools Customizer for z/OS 1.1)

Table 1. Steps for customizing Object Comparison Tool for the first time (continued)

Step	Description	Instructions
Create Db2 entries.	1. On the **Customizer Workplace (CCQPWRK)** panel, issue the ASSOCIATE primary command, and press Enter. 2. On the **Associate DB2 Entry for Product (CCQPDAD)** panel, issue the CREATE primary command, and press Enter. 3. On the **Create DB2 Entries (CCQPCDB)** panel, specify the information for the new Db2 entry, and press Enter. 4. On the **Associate DB2 Entry for Product (CCQPDAD)** panel, issue the A line command against the new Db2 entry, and press Enter. Create new Db2 entries and associate them with Object Comparison Tool.	Creating and associating DB2 entries (IBM Tools Customizer for z/OS 1.1)
Define product parameters.	1. On the **Customizer Workplace (CCQPWRK)** panel, specify the E line command against the **Product parameters** field. 2. On the **Product Parameters: DB2 Object Comparison (CCQPPRD)** panel, specify your parameter values. Required parameters are indicated by an asterisk (*). 3. Press Enter to save and exit.	Defining product or component parameters (IBM Tools Customizer for z/OS 1.1)
Generate the jobs.	On the **Customizer Workplace (CCQPWRK)** panel, issue the G line command against the new Db2 entry, and press Enter.	Generating customization jobs (IBM Tools Customizer for z/OS 1.1)
Optional: Edit the jobs.	Ensure that the GOCFB2VB job contains the correct ADB and GOC SAMP data sets. If not, edit this job to correct those values.	“Editing the GOCFB2VB job” on page 22
Submit the jobs.	On the **Finish Product Customization (CCQPCST)** panel, issue the E line command against the *ab*CUST*xy* member.	“Submitting the customization jobs” on page 24
Propagate the customizations to additional LPARs as needed.	If you have a multiple-LPAR environment, use one of the specified methods to propagate your customization to other LPARs.	“Using Tools Customizer in a multiple-LPAR environment” on page 21

Roadmap: Recustomizing Db2 Object Comparison Tool

After you have initially customized Db2 Object Comparison Tool by using Tools Customizer (TCz), you might later need to recustomize it to change one or more parameter values. For example, when you apply maintenance, the instructions might direct you to recustomize Object Comparison Tool.

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, or both. Use the method in this roadmap instead of deleting customization jobs from the customization library.

To recustomize Object Comparison Tool, complete the steps in the following table.

Tips:

- For multiple-LPAR environments, determine your customization strategy first: "Using Tools Customizer in a multiple-LPAR environment" on page 21.
- Use a new customization library every time that you apply maintenance and regenerate all the TCz jobs (by using the GENERATEALL command). For example, append a date as show in the following example:

```
Customization lib: RSTEST.AOC.$RS01$.ADB1210.D200716
```

  This practice provides a backup and allows you to compare the jobs to a previous customization by using ISPF option 3.12.
- For guidance on any input fields in TCz, position your cursor on the input field and press F1 (Help).

Table 2. Required steps for recustomizing Object Comparison Tool

Step	Description	Instructions
Start TCz.	1. On the ISPF Command shell panel, issue the following command: `EX 'hlq.TCZ110.SCCQEXEC(CCQTCZ)'`	Starting Tools Customizer (IBM Tools Customizer for z/OS 1.1)
Specify the Object Comparison Tool metadata library.	1. On the **IBM Tools Customizer for z/OS (CCQPHME)** panel, specify option 1 (**Customize a product**). 2. On the **Specify the Product or Pack Metadata Library (CCQPHLQ)** panel, enter the following value in the **Product or pack metadata library** field, and press Enter: `DMTOOL.SGOCDENU`	Specifying the metadata library for the product or pack to customize (IBM Tools Customizer for z/OS 1.1)
Define product parameters.	1. On the **Customizer Workplace (CCQPWRK)** panel, specify the E line command against the **Product parameters** field, and press Enter. 2. Edit the specific tasks, steps, or parameters that you want to change. 3. Press Enter to save and exit.	Defining product or component parameters (IBM Tools Customizer for z/OS 1.1)
Generate the jobs	On the **Customizer Workplace (CCQPWRK)** panel, issue the G line command against a site-specific SSID, and press Enter.	Generating customization jobs (IBM Tools Customizer for z/OS 1.1)

Table 2. Required steps for recustomizing Object Comparison Tool (continued)

Step	Description	Instructions
Optional: Edit the jobs.	Ensure that the GOCFB2VB job contains the correct ADB and GOC SAMP data sets. If not, edit this job to correct those values.	"Editing the GOCFB2VB job" on page 22
Submit the jobs.	On the **Finish Product Customization (CCQPCST)** panel, submit the generated jobs in the order they are displayed.	"Submitting the customization jobs" on page 24
Propagate the customizations to additional LPARs as needed.	If you have a multiple-LPAR environment, use one of the specified methods to propagate your customization to other LPARs.	"Using Tools Customizer in a multiple-LPAR environment" on page 21

Using Tools Customizer in a multiple-LPAR environment

Tools Customizer (TCz) supports customizations on only the local LPAR. However, you can propagate customizations to additional LPARs.

About this task

In a multiple-LPAR environment, TCz identifies the LPAR to which you are logged on and uses this LPAR name for several parameter settings, including the data store. Therefore, you can use the TCz data store to customize only that LPAR.

Procedure

To customize products that run against Db2 subsystems on multiple LPARs, use one of the following methods:

- **Method 1: Customize a single Db2 subsystem or data sharing group and copy the customization jobs to each LPAR**
  a) Customize one Db2 subsystem or member.
  For example, you might customize member DB1S in group DBGS in your sandbox environment.
  b) If you are using data sharing, propagate that customization to the other members in the group:
    a. Copy the customization jobs to the other members.
    For example, copy the jobs for DB1S to member DB2S.
    b. Edit the jobs as needed for the subsystem and LPAR.
    For example, replace the member names. Depending on your environment, you might also need to replace data set names. You can use a REXX exec to do this customization.
    c. Run those jobs.
    Some jobs do not need to be run on every member in a group. Some jobs only need to run once per LPAR or Sysplex. To determine where a job needs to be run, look at the job listings on the **Finish Product Customization (CCQPCST)** panel. Depending on the values of the **SSID** and **GrpAttch** columns, take the following actions for each job:

Table 3.

SSID column value	GrpAttch column value	Action	Comments
--	--	Run once per LPAR	None
--	A group name	Run once per group	None

Table 3. (continued)			
SSID column value	**GrpAttch column value**	**Action**	**Comments**
A member name	A group name	Run once per member in the group	None
An SSID	--	Run once	This entry is for a stand-alone Db2 subsystem.

c) Copy the jobs from the initial customized subsystem or member to all of your other subsystems or groups. Then, edit those jobs, preferably with a REXX exec, and run them.

For example, copy the jobs for DB1S in group DBGS to the members DB1D and DB2D in your development group DBGD, edit those jobs as needed, and run them. Then, copy the jobs for DB1S to the members DB1T and DB2T in your test group DBGT, edit those jobs, and run them. Continue until all groups are customized.

- **Method 2: Generate customization jobs for each Db2 subsystem and copy those jobs to the appropriate LPARs**

a) Associate all Db2 entries in one instance of TCz on one LPAR, regardless of the LPARs on which the Db2 subsystem resides.

b) Generate customization jobs for each Db2 entry.

c) Copy the generated customization jobs to the LPAR to run against the specific Db2 entries. You might need to edit these customization jobs for specific LPARs. For example, you might need to edit the data set names. (Otherwise, you generally do not need to make manual changes to the jobs that are customized by TCz.)

Editing the GOCFB2VB job

The GOCFB2VB job is generated by TCz if you specified that you wanted to create variable-blocked (VB) versions of the Db2 Admin Tool and Object Comparison Tool CLIST and EXEC libraries You might need to edit this job to specify the correct ADB and GOC SAMP data sets.

About this task

GOCFB2VB is generated based on the information specified on the **Product Parameters: DB2 Object Comparison (CCQPPRD)** panel under the following field:

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

GOCFB2VB is based on the GOCFB2VB template and is in member *job_sequence_number*_FB2V *_Db2_entry_ID*.

The following parameters in GOCFB2VB correspond to the indicated field on **Product Parameters: DB2 Object Comparison (CCQPPRD)** panel:

Table 4. GOCFB2VB parameters	
Parameter name	**Field**
CCQ_GOC_HLQ	**DB2 Object Comparison hlq**
CCQ_GOC_FB2VB_VLSRNM	**Fixed to variable blocked VOLSER**
CCQ_GOC_FB2VB_DASD	**Fixed to variable blocked UNIT**
CCQ_GOC_ADB_HLQ	**DB2 Admin Tool hlq**

Procedure

To edit the GOCFB2VB job:

1. Open the GOCFB2VB job in the ISPF editor.
2. Edit the job step that creates the VB version of the CLIST library. Check the low-level qualifier for the VB data set and correct it if needed.

   For example, in the following job step, SGOCCLST is specified as the low-level qualifier:

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

   Figure 2. Example GOCFB2VB job step that creates the VB version of the CLIST library

   This job creates the GOCC10.*low-level-qualifier*.VB data set, where *low-level qualifier* is the low-level qualifier that you specify.
3. Edit the job step that creates the VB version of the EXEC library. Check the low-level qualifier for the VB data set and correct it if needed.

   For example, in the following job step, SGOCEXEC is specified as the low-level qualifier:

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

   Figure 3. Example GOCFB2VB job step that creates the VB version of the EXEC library

   This job creates the GOCC10.*low-level-qualifier*.VB data set, where *low-level qualifier* is the low-level qualifier that you specify.
4. Save the file.

What to do next

Submit the job.

Submitting the customization jobs

After TCz generates the customization jobs for Db2 Object Comparison Tool, you must submit them to complete the customization process. SYSADM or equivalent authority is required to run the generated jobs.

About this task

TCz 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 TCz generates.

Table 5. Customization jobs that TCz 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

The following figure shows part of the **Finish Product Customization (CCQPCST)** panel. The table on this panel shows the customization jobs that are generated by TCz. They are grouped by job sequence number.

```
CCQPCST                 Finish Product Customization          Row 1 to 2 of 2
 Command ===>                                                 Scroll ===> PAGE

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


 Line Commands: E - Edit  B - Browse

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

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

*Figure 4. The **Finish Product Customization (CCQPCST)** panel*

The member-naming conventions depend on whether the customization jobs are for Db2 entries, an LPAR, or the product, as follows:

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 TCz 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. Object Comparison Tool assigns the template name.

DB2_entry_ID
: Two alphanumeric characters, AA - 99, that TCz assigns to a Db2 entry.

For example, the XYZBND*DB2_entry_ID_1* and XYZBND*DB2_entry_ID_2* jobs are generated from the XYZBNDGR template, and the XYZ4*DB2_entry_ID_1* and XYZ4*DB2_entry_ID_2* jobs are generated from the XYZ4 template. If the jobs are generated on two Db2 entries, the following member names are listed sequentially: A0BNDGAA, A0BNDGAB, A14AA, A14AB.

Customization jobs for an LPAR or the product
The members use the following naming convention:

```
<job_sequence_number><job_ID>
```

where

job_sequence_number
Two alphanumeric characters, A0 - Z9, that TCz 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. 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.

Use the **New** column to determine whether the job member needs to be submitted:

YES
The job member is newly created or updated and needs to be submitted for customization.

NO
The job member is not newly created or updated and does not need to be submitted for customization.

Procedure

Submit the generated customization jobs by following the process that you use in your environment or by using the following method:

1. Specify B or E against a customization job or the product customization library, and press Enter.
   An ISPF browsing or editing session is started.
2. Browse the customization job or each member in the library to ensure that the information is correct.
3. Run the TSO `SUBMIT` command.
4. Press End.

Results

Object Comparison Tool is customized, and the **Customizer Workplace (CCQPWRK)** panel is displayed. For the Db2 entries on which Object Comparison Tool was customized, the status is 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 Object Comparison Tool.

Allocating libraries for Db2 Object Comparison Tool

Before you can use Db2 Object Comparison Tool, you must first allocate the libraries to your ISPF session.

Procedure

- To allocate the Object Comparison Tool libraries to your ISPF session, choose one of the following three methods that is most appropriate for your installation:
  - Use the PRODADD and LIBAPRE parameters on the ADBL CLIST to specify the unique library names for Object Comparison Tool libraries.

    If you are currently using the ADBL CLIST to allocate the Db2 Administration Tool ISPF libraries (by using the LIBDEF service), you should also use the ADBL CLIST to allocate the 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, copy Object Comparison Tool ISPF libraries into these same libraries or allocate additional ISPF data sets.
  - If you have a personal set of ISPF libraries, copy the Object Comparison Tool ISPF libraries to these data sets. To verify that you have allocated (using 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.
- If you plan to run compare jobs online, also ensure that the Db2 libraries are set up properly.

  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.
  ```

Related concepts

“ADBL CLIST for invoking Db2 Object Comparison Tool” on page 30
The ADBL CLIST, in the SADBCLST library, is provided for running Db2 Admin Tool or Object Comparison Tool.

Customizing the JCL that Object Comparison Tool uses

You might need to customize the Object Comparison Tool JCL to adhere to your installation standards. You can configure the JCL that is used by Object Comparison Tool to run Db2 utilities and other Db2 functions by modifying the skeletons in the SADBSLIB and SGOCSLIB libraries. Most other skeletons will not require configuration.

Procedure

To customize the JCL that Object Comparison Tool uses:

- Configure the following members of SADBSLIB as needed:

  ADBAPY
  Generates an apply job or step (uses ISPF batch)

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

  ADBDCMD
  Executes Db2 commands

ADBEDDL
Executes DDL files (DROP, CREATE, ALTER)

ADBTCHK
Generates a CHECK DATA job or step

ADBTHPU
Generates a High Performance Unload job or step

ADBTIMC
Generates an image copy job or step

ADBTREL
Generates a LOAD or RELOAD job or step

ADBTREO
Generates a REORG job or step

ADBTREN
Generate a RUNSTATS job or step

ADBTUNL
Generates an UNLOAD job or step

ADBS27AC
Generates a convert job or step

- Configure the following members of SGOCSLIB as needed:

GOCCMP
Generates a compare job or step

GOCDB2
Generates extractions from the Db2 catalog for the source or target

GOCDDL
Generates extractions from the DDL for the source or target

Customizing data set names

You can set up Db2 Admin Tool to use your local naming conventions for data sets.

Procedure

To customize data set names, modify the ADB2UCUS skeleton that resides in the ISPSLIB library as follows:

- Edit any data set names that are preceded by SET statements as needed. (SET statements are indicated by) SET.)
- Use variables for the data set names as needed. A complete list of variables is included in the SLIB member ADB2UCUT. Some of the variables you can use are:

&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)

- Ensure that data set names do not extend beyond column 71 in the ADB2UCUS data set. Any characters beyond column 71 are truncated.
- Ensure that generated data set names, including periods, will not be longer than 44 bytes.

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.

Tip: 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). 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 SADBSAMP library. Instructions for completing this step are provided in sample job ADBU002.

Example

This example demonstrates several different types of changes to the variable ASYCPY1.

The variable ASYCPY1 is shipped as follows:

```
)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
```

Enabling product discovery for Object Comparison Tool

Product discovery is the ability of one product to determine whether another product is installed without invoking that second product. For example, IBM Db2 Administration Foundation for z/OS can determine, or *discover*, whether Object Comparison Tool is installed without invoking the object compare options in Db2 Administration Tool.

Enabling product discovery for Object Comparison Tool is optional but strongly recommended if Db2 Administration Foundation is also installed. Enabling this product discovery allows you to use certain additional functions in Db2 Administration Foundation.

Before you begin

The PTF for APAR PH55178 must be installed.

About this task

The following files are used for product discovery. Both files are provided as members of the SGOCSAMP data set and are in YAML format:

GOCDSCVP
The product file, which contains basic information about the product ID and release. Do not edit this file.

GOCDSCVS
The base for the customization file, which contains the names of the installation target library data sets and the location of the corresponding product file. You will edit this file to provide this information as part of the following procedure.

Note: Any future updates to GOCDSCVP or GOCDSCVS will be indicated in the ++HOLD information for the relevant PTF that updates the file. For example, GOCDSCVP might be updated if additional product features need to be discovered.

Procedure

To enable product discovery for Object Comparison Tool:

1. Copy GOCDSCVP and GOCDSCVS in the SGOCSAMP library to another location where they can be accessed by products discovering Object Comparison Tool.

   Copying these files ensures that maintenance to the files does not result in unexpected updates to the execution environment.

   You can copy these files to sequential data sets, members of a PDS or PDSE, or UNIX System Services files. Name these files to conform to your installation naming standards, but also consider the following recommendations.

   Naming recommendations:

   - If the files are stored in sequential data sets, use the low-level qualifiers PROD (product file) and CUST (customization file).

     Example:
     `DB2TOOLS.GOC131.PROD` — a copy of the GOCDSCVP product file
     `DB2TOOLS.GOC131.CUST` — a customized version of the GOCDSCVS file

   - If the files are stored in a PDS or PDSE, base the member names on the Object Comparison Tool release number and the type of file. Including the release in the member name allows you to retain files for multiple versions if your installation has multiple versions in use.

     Example:
     `DB2TOOLS.GOC.DISCOVER(GOC131P)` — a copy of the GOCDSCVP product file
     `DB2TOOLS.GOC.DISCOVER(GOC131C)` — a customized version of the GOCDSCVS file

   - If the files are stored in the UNIX System Services file system, use the following paths:

<optional prefix>/usr/lpp/db2tools/goc/prod/goc131P.yaml — a copy of the GOCDSCVP product file
<optional prefix>/usr/lpp/db2tools/goc/site/goc131C.yaml — a customized version of the GOCDSCVS file

2. Edit the copy of GOCDSCVS as described in the file comments.
3. If Zowe-based products that use IBMUnified Management Server for z/OS (UMS), such as Db2 Administration Foundation, will be used with Object Comparison Tool, update the UMS parameters to indicate the location of the product and customization files. For details, see Unified Management Server for z/OS 1.2 documentation.

ADBL CLIST for invoking Db2 Object Comparison Tool

The ADBL CLIST, in the SADBCLST library, is provided for running Db2 Admin Tool or Object Comparison Tool.

The ADBL CLIST opens the Db2 Admin Tool main menu. Use the PANEL(GOCMENU) parameter to instead bring up the Db2 Object Comparison Tool 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 searches only the CLIST libraries.

Related information

Invoking Db2 Admin Tool (IBM Db2 Administration Tool for z/OS 13.1.0)

Chapter 3. Getting started with Db2 Object Comparison Tool

Object Comparison Tool runs as an extension to Db2 Admin Tool. The ISPF full-screen interface uses Db2 Admin Tool functions to display panels and run SQL statements.

As part of the panel interface, Object Comparison Tool provides a walk-through option that leads you through the process of creating a job to compare Db2 objects. This end-to-end framework guides you through the options that you need to specify. ISPF help panels are also available Object Comparison Tool. To display a help panel, enter HELP or press PF1.

Opening Object Comparison Tool

You invoke Object Comparison Tool from the main menu in Db2 Admin Tool.

Procedure

On the **DB2 Administration Menu (ADB2)** panel, specify option C, and press Enter:

```
ADB2 dmin -------------- DB2 Administration Menu 13.1.0 ----------------- 17:50
Option ===> C

   1 - DB2 system catalog                                DB2 System: DD1A
   2 - Execute SQL statements                            DB2 SQL ID: ADM001
   3 - DB2 performance queries                           Userid    : ADM001
   4 - Change current SQL ID                             DB2 Schema: ADM001
   5 - Utility generation using LISTDEFs and TEMPLATEs   DB2 Rel   : 1315
   P - Change DB2 Admin parameters                       DB2 F.Lvl : V13R1M501
  DD - Distributed DB2 systems                           Max ApplC : V13R1M500
   E - Explain                                           ApplCompat: V13R1M500
   Z - DB2 system administration                         Cat Level : V13R1M501
  SM - Space management functions
   W - Manage work statement lists
   X - Exit DB2 Admin
  CC - DB2 catalog copy version maintenance
  CM - Change management
                                                                More:     +
Interface to other DB2 products and offerings:
   I  DB2I
   C  DB2 OBJECT COMPARISON TOOL
```

*Figure 5. **DB2 Administration Menu (ADB2)** panel*

The **DB2 Object Comparison Tool Menu (GOCMENU)** panel is displayed:

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

                                        Specification Status:
   1  - Specify compare source (new)     Incomplete
   2  - Specify compare target (old)     Incomplete
   3  - Specify compare masks            None specified
   4  - Specify ignores                  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
   MR- Manage saved compare results
```

*Figure 6. **DB2 Object Comparison Tool Menu (GOCMENU)** panel*

Related reference

"Object Comparison Tool main menu" on page 32

Use the **DB2 Object Comparison Tool Menu (GOCMENU)** panel to specify the criteria for the comparison that you want to run.

Object Comparison Tool main menu

Use the **DB2 Object Comparison Tool Menu (GOCMENU)** panel to specify the criteria for the comparison that you want to run.

```
GOCMENU ----------- DB2 Object Comparison Tool Menu 13.1.0 -------------- 10:05
Option ===>

                                       Specification Status:
   1  - Specify compare source (new)     Incomplete
   2  - Specify compare target (old)     Incomplete
   3  - Specify compare masks            None specified
   4  - Specify ignores                  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
   MR - Manage saved compare results
```

*Figure 7. **DB2 Object Comparison Tool Menu (GOCMENU)** panel*

This panel has the following options:

1 - Specify compare source (new)
Select this option to begin specifying the Db2 source objects to be compared. For detailed instructions, see "1. Specifying source objects" on page 48.

2 - Specify compare target (old)
Select this option to begin specifying the Db2 target objects to be compared. For detailed instructions, see "2. Specifying target objects" on page 60.

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. For detailed instructions, see “3. Specifying compare masks” on page 63.

4 - Specify fields to ignore
Select this option to specify that certain fields should be ignored when the comparison is performed. For detailed instructions, see “4. Specifying ignores” on page 73.

5 - Generate compare job
Select this option to generate the batch compare job. For detailed instructions, see “5. Generating a compare batch job” on page 78.

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 (GOCMENU)** panel and selecting them individually.

V - Generate job to extract version file from source only
Generates a batch job that creates a version file from the source only. This version file can be used for the source or target in subsequent compare operations. This option can be used to create a version file on one system, transfer the version file to another system, and then generate a compare job on the other system.

R - Reset all
Clears the **Specification Status** fields for all options. You can then enter new specifications for each option.

RS - Reset Source
Clears the **Specification Status** field for option 1. You can then specify a new source.

RT - Reset Target
Clears the **Specification Status** field for option 2. You can then specify a new target.

S - Save dialog
Stores the current selections for later retrieval and subsequent reuse. For information about how to save a dialog, see “Saving dialogs” on page 110.

M - Manage/Restore dialog
Select this option to retrieve, rename, or delete a previously saved dialog. For information about how to use these functions, see “Managing and restoring dialogs” on page 111.

MC - MultiCompare
Select this option to compare one or more saved dialogs. For information about comparing multiple objects, see Chapter 8, “Comparing multiple sources and targets,” on page 131.

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

Db2 Object Comparison Tool scenarios

The following common scenarios illustrate how to use Db2 Object Comparison Tool.

Scenario: Comparing a Db2 development catalog to a Db2 production catalog

When you make changes on your development system, such as creating a new table or view or changing an existing table, you might want to eventually make those same changes on your production system. To do so, you can use Db2 Object Comparison Tool to compare your development catalog to your production catalog. Then, Object Comparison Tool can make changes in the production catalog so that the objects in both systems are the same.

Procedure

To compare a Db2 development catalog to a Db2 production catalog:

1. Specify the source (your development catalog):

a) On the **DB2 Object Comparison Tool Menu (GOCMENU)** panel, select option **1 - Specify compare source (new)**, and press Enter.

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

b) On the **Specify Compare Source (GOC1)** panel, select option **2 - Source is from the DB2 catalog**, and press Enter.

c) On the **Specify DB2 Source Catalog Extract (GOC12)** panel, complete the following fields:

Data set name
The name of the data set that you want to use for the version file for the source, such as `devdb.v23.D080319`. (Version files are created as part of the compare process. These files store information about the objects to be compared.)

Tips:

- Save all of your version files for future comparisons and the ability to undo changes at a later time if needed.
- Plan a naming convention to help keep track of the version files and easily find them. One possible naming convention is to include the date, as in the preceding example (`D080319`).

Description
A description of the source, such as `development database`.

Tip: For this scenario, the description is simple. When you are doing your own comparisons, assign descriptive names to your version files so that you can easily find them, and include the date that they were created. For example:

```
Accounting V9 R10 M08 2019-04-01
```

d) Select option **1 - Source is databases from the DB2 catalog**, and press Enter.

e) On the **Specify Source DB2 Databases (GOC1D)** panel, use the `I` line command to insert a line.

f) On the **Compare Add Databases (GOC1DA)** panel, specify the source database or databases by entering a partial database name and pressing Enter.

For example, if you enter `AGBL` in the **Partial database name** field, all databases that begin with AGBL are displayed.

g) On the **Compare Add Databases (GOC1DD)** panel, use the S line command to select the specific databases that you want to compare.

h) Exit back to the **DB2 Object Comparison Tool Menu (GOCMENU)** panel. Notice that the **Specification Status:** next to option **1** is now **Database extract specified**.

2. Specify the target (your production catalog):

a) Select option **2 - Specify compare target (old)**, and press Enter.

b) On the **Specify Compare Target (GOC1)** panel, select option **2 - Target is from the DB2 catalog**, and press Enter.

c) On the **Specify DB2 Target Catalog Extract (GOC12)** panel, complete the following fields:

Data set name
The name of the data set to use for the version file for the target, such as `proddb.v23.D080311`.

Description
A description of the target, such as `production database scenario`.

d) Select option **1 - Target is databases from the DB2 catalog**, and press Enter.

e) On the **Specify Target DB2 Databases (GOC1D)** panel, use the `I` line command to insert a line.

f) On the **Compare Add Databases (GOC1DA)** panel, specify the database that contains the target by entering a partial database name (such as `DGWD`) and a location name (such as `STLEC1`), and press Enter.

g) On the **Compare Add Databases (GOC1DD)** panel, use the **S** line command to select the target database or databases that you want to compare with the source.

h) Exit back to the **DB2 Object Comparison Tool Menu (GOCMENU)** panel. Notice that the **Specification Status:** next to option **2** is now **Database extract specified**.

3. Specify any compare masks:

   Often, the names of objects in your development system are not the same as the names in your production system. Even if the names are the same, the owner IDs might be different. You can use compare masks to account for these differences. Db2 Object Comparison Tool can then match the appropriate objects for the comparison, even if the names are different.

   a) On the **DB2 Object Comparison Tool Menu (GOCMENU)** panel, select option **3 - Specify Compare Masks**, and press Enter.

   b) On the **Specify Compare Masks (GOC3)** panel , complete the following fields, and press Enter:

      Mask DSN
      The name of the data set for the masks. If the data set does not exist, it is created.

      Edit Mask
      YES

      This scenario shows you how to define masks in a data set. Alternatively, if Change Management is enabled, you can define masks in the Change Management repository.

   c) On the **Edit Compare Masks (GOCEDIT)** panel, insert a line for each mask.

      For example, the following lines define name masks:

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

      For information about mask definitions and syntax, see “Translation masks” on page 65.

      For example, `OWNER: ABC*,DEF*` specifies that all owner names of ABC* in the source are translated to DEF* for the comparison. (The asterisk is a wild card.) In this case, ABCDBA in the source matches DEFDBA in the target.

      Tip: Usually, the compare process is iterative. You generate a compare job and then analyze the differences in the report to see what masks you need to create for the next run of the compare job.

   d) Issue the SAVE command

   e) Exit back to the **DB2 Object Comparison Tool Menu (GOCMENU)** panel. Notice that the **Specification Status:** next to option **3** is now **Mask specified**.

      The next step that is listed on the **DB2 Object Comparison Tool Menu (GOCMENU)** panel is to specify ignore fields (option **4 - Specify ignores**). Ignore fields are characteristics that you want to be ignored during the comparison. For example, different buffer pool names in the source and target might be acceptable, and you do not want this difference to result in a change. At this point, assume that you do not know of any such differences that you want to ignore. So skip this step for now.

4. Generate and run a compare job:

   a) On the **DB2 Object Comparison Tool Menu (GOCMENU)** panel, select option **5 - Generate compare job**, and press Enter.

   b) On the **Generate Compare Jobs (GOC5)** panel, specify the following options, and press Enter:

      Worklist name
      TEST

      Suppress DROP of objects
      YES

PDS for jobs
TEST

Prefix for data sets
TEST

Single compare job
YES

Member name
COMPARE

Set all of the remaining options to NO, N, or blank.

Notice that for this first comparison, you are not requesting that any apply jobs be generated. Typically, you want to look at the comparison report first and make any changes before you generate apply jobs.

Tip: The settings for parameters are persistent; they settings that you specified previously remain until you change them.

c) In the generated JCL job that is displayed, make any changes to the JCL as needed. For example, you might need to change the JOB statement.

d) Type the SUB command, and press Enter to submit the job.

e) Check that the job completed successfully.

5. Check the report to see the differences between the source and target:

a) In the job output, look at the information under the line `OBJECT COMPARISON REPORT`.

This report shows the differences between the source and target objects. It lists the differences as changes that need to be made to the target so that it matches the source.

For this scenario, suppose that you notice the following items in the output:

```
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 'STOGROUP AGBLPSG' TO 'STOGROUP AGBLTSG'
  Tablespace will be altered
```

The CLOSE attribute, PRIQTY attribute, and STOGROUP name are all listed as changed. The preceding lines in the output mean that the values are different in the source and target. However, in this case, suppose that you do not want to change the name of STOGROUP or the values of the CLOSE and PRIQTY attributes. Therefore, you need to set a mask for STOGROUP and ignore fields for CLOSE and PRIQTY and then run a comparison job again.

Suppose that you also notice in the report that objects are altered, dropped, and added:

```
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 this scenario, assume that these changes are changes that you want to make to your production system.

6. Add the additional mask and ignore fields.

a) On the **DB2 Object Comparison Tool Menu (GOCMENU)** panel, select option **3 - Specify Compare Masks**, and press Enter.

   The **Specify Compare Masks (GOC3)** panel should list the same data set name that you originally specified (in step "3" on page 35).

b) Make sure the **Edit Mask** field is still set to YES, and press Enter.

c) On the **Edit Compare Masks (GOCEDIT)** panel, add `SGNAME: *TSG,*PSG`.

   (This mask accounts for the difference in the STOGROUP names AGBLPSG and AGBLTSG.)

d) Exit back to **DB2 Object Comparison Tool Menu (GOCMENU)** panel, select option **4 - Specify ignores**, and press Enter.

e) On the **Specify Compare Ignores (GOC4)** panel, specify the following values, and press Enter:

   Data Set Name
   The name of a data set for the ignore file, such as `IGNORE.DATA`

   Edit Ignore Fields Specification
   YES

f) On the **Specify Ignore Fields : Objects (GOCCI)** panel, use the **U** line command to update the SYSTABLESPACE object, and press Enter.

g) In the **Specify Ignore Fields for object (GOCCIF)** panel, use the **S** line command to select CLOSERULE and PQTY.

   Ignore fields are specified according to columns in the Db2 catalog. In this case, you want to ignore the CLOSE and PRIQTY attributes of the table space. Those values are captured in the CLOSERULE and PQTY columns of SYSIBM.SYSTABLESPACE. Therefore, CLOSERULE and PQTY need to be selected.

h) Press PF3 to exit.

   On the **Specify Ignore Fields : Objects (GOCCI)** panel, CLOSERULE and PQTY are listed in the **Ignore Fields** column for SYSTABLESPACE.

i) Exit back to the **DB2 Object Comparison Tool Menu (GOCMENU)** panel.

   Notice that the **Specification Status:** next to option **4** is now **Ignore fields specified**.

7. Generate another compare job with the new mask and ignore fields:

   Db2 Object Comparison Tool created version files during the first comparison operation. You can now use these version files instead of choosing the objects from the catalog. Because version files are compressed to save space, using them can save you both time and CPU.

a) On the **DB2 Object Comparison Tool Menu (GOCMENU)** panel, select option **1 - Specify compare source (new)**, and press Enter.

b) This time, on the **Specify Compare Source (GOC1)** panel, specify option **3 - Source is from a compare version file**, and press Enter.

c) On the **Specify Source Compare Version File (GOC13)** panel, specify the name of the data set that contains the version file for the source, and press Enter.

   This data set name is the one you specified in step "1" on page 33.

d) On the **DB2 Object Comparison Tool Menu (GOCMENU)** panel, select option **2 - Specify compare target (old)**, and press Enter.

e) This time, on the **Specify Compare Target (GOC1)** panel, select option **3 - Target is from a compare version file**, and press Enter.

f) On the **Specify Target Compare Version File (GOC13)** panel, specify the name of the data set that contains the version file for the target, and press Enter.

   This data set name is the one you specified in step "2" on page 34.

   After you press Enter, notice that on the **DB2 Object Comparison Tool Menu (GOCMENU)** panel, the **Specification Status:** next to options **1** and **2** is now **Compare version file specified**.

g) Select option **5 - Generate compare job**, press Enter, and complete the steps that you did before to generate and run the job.

h) Check the report output.

You should see the mask and ignore fields that you specified.

8. Apply the changes to synchronize your production and development systems:

a) On the **DB2 Object Comparison Tool Menu (GOCMENU)** panel, select option **5 - Generate compare job** again, and press Enter.

b) This time, on the **Generate Compare Jobs (GOC5)** panel, set the **Generate Apply Job** field to Yes and set any other fields as needed, and press Enter.

c) If the **Change Management Prompt (ADB2CMRO)** panel opens, specify NO.

(For this scenario, assume that you are not using Change Management.)

d) On the **Specify Data Set Name for Apply Jobs (GOC5AJ)** panel, enter the name of a data set where you want the apply jobs generated.

e) Edit the generated comparison job as needed, and submit the job.

f) Check the output to confirm that the job completed successfully.

g) Run the generated apply job to make the changes to your production catalog.

Scenario: Undoing changes that were made in a catalog-to-catalog comparison

Suppose that you used Object Comparison Tool to compare two Db2 catalogs and then apply changes to the target catalog so that it matches the source catalog. Later, you decide that you do not want those changes. Db2 Object Comparison Tool can undo those changes for you.

Before you begin

To undo the changes, you need the version files from the catalog-to-catalog comparison.

About this task

Assume that you want to undo the changes that you made in "Scenario: Comparing a Db2 development catalog to a Db2 production catalog" on page 33 and restore the target (the production catalog) to its state prior to the comparison.

Important: Any data that is added between the time that the compare synchronization is done (step "3" on page 39) and the time that the undo changes process is done (step "4" on page 39) might be lost.

Procedure

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

1. Specify the compare source.

In this scenario, the source is the version file for the target in the original comparison. This version file represents the production catalog prior to the changes.

a) On the **DB2 Object Comparison Tool Menu (GOCMENU)** panel, select option **1 - Specify compare source (new)**.

b) On the **Specify Compare Source (GOC1)** panel, specify option **3 - Source is from a compare version file**.

c) On the **Specify Source Compare Version File (GOC13)** panel, specify the name of the data set that contains the version file, and press Enter.

(This data set name is the name that you specified in step "2" on page 34 in "Scenario: Comparing a Db2 development catalog to a Db2 production catalog" on page 33.)

On the **DB2 Object Comparison Tool Menu (GOCMENU)** panel, notice that the **Specification Status:** next to option **1** is now **Compare version file specified**.

2. Specify the compare target.

   In this scenario, you want Object Comparison Tool to determine the target objects from the current production catalog based on the source version file.

   a) On the **DB2 Object Comparison Tool Menu (GOCMENU)** panel, select option **2 - Specify compare target (old)**.

   b) On the **Specify Compare Target (GOC1)** panel, select option **4 - Target is from the DB2 catalog and objects are automatically selected based on the selected source objects**.

   c) On the **Specify Target DB2 Location (GOC14)** panel, enter the following information, and press Enter:

   - The location of your production subsystem.
   - The name of a data set to use for the target version file. If the data set does not already exist, it is created.

   On the **DB2 Object Comparison Tool Menu (GOCMENU)** panel, notice that the **Specification Status:** next to option **2** is now **Automatic (DB2 catalog extract)**.

   For this scenario, do not specify any masks or ignore fields.

3. Generate the compare job:

   a) On the **DB2 Object Comparison Tool Menu (GOCMENU)** panel, select option **5 - Generate compare job**.

   b) On the **Generate Compare Jobs (GOC5)** panel, specify the following options, and press Enter:

   Worklist name
   TEST

   Suppress DROP of objects
   YES

   PDS for jobs
   TEST

   Prefix for data sets
   TEST

   Single compare job
   YES

   Member name
   COMPARE

   Set all the remaining options to NO or N.

   c) Edit the generated compare job as needed, and submit the job.

   d) Check the output to confirm that the job completed successfully.

   e) Check the compare report to make sure that the expected changes are listed.

4. Generate the apply job to undo the changes that you made previously in the catalog-to-catalog comparison:

   a) On the **DB2 Object Comparison Tool Menu (GOCMENU)** panel, select option **5 - Generate compare job**.

   b) On the **Generate Compare Jobs (GOC5)** panel, specify the following additional options, and press Enter:

   Generate apply jobs
   YES

   Generate one job
   YES

   Member prefix
   APPLY

Content of apply job(s)
ALL

Unload method
U

IDENTITY START value
ORIGINAL

Run REORG/REBUILD
A

c) If the **Change Management Prompt (ADB2CMRO)** panel opens, specify N0.

(For this scenario, assume that you are not using Change Management.)

d) On the **Specify Data Set Name for Apply Jobs (GOC5AJ)** panel, enter the name of the data set where you want the apply job generated.

e) Edit the generated compare job as needed, and submit the job.

f) Check the output to confirm that the job completed successfully.

g) Run the generated apply job to restore the production catalog to the state before the changes were made.

Scenario: Comparing DDL to a catalog

You can compare the DDL for a single object to the Db2 catalog to make changes on the system for only that object.

About this task

Suppose you want 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. So, you generate DDL that shows how the table will look after the change. The DDL will be your source for the comparison. It contains only a CREATE TABLE statement for the table. The DDL does not include any related indexes, foreign keys, or other related objects. Those objects will not be changed, because they are not included in the source DDL.

You then specify that the compare target be selected automatically from the Db2 catalog. In this case, Db2 Object Comparison Tool determines how to change the table. If the table does not currently exist in the target, Object Comparison Tool creates the table. If the table exists, Object Comparison Tool uses the version file instead of the catalog. Object Comparison Tool might alter the table or drop and re-create it, depending on the changes that need to be made. Db2 Object Comparison Tool restores objects and dependencies, such as indexes. If the table needs to be dropped and re-created, Db2 Object Comparison Tool also re-creates objects that have been dropped as a result of dropping the table. The table data is unloaded and, after the object definitions are applied, reloaded back into the table.

Procedure

To compare DDL to a catalog:

1. Specify the compare source.

   In this scenario, the source is the table definition in the DDL.

   a) On the **DB2 Object Comparison Tool Menu (GOCMENU)** panel, select option **1 - Specify compare source (new)**, and press Enter.

   b) On the **Specify Compare Source (GOC1)** panel, select option **1 - Source is from a DDL file**, and press Enter.

   c) On the **Specify Source DDL File (GOC11)** panel, specify the following information, and press Enter:

   - The name of the data set that contains the DDL.
   - The name of a data set to use for the source version file.

   On the **DB2 Object Comparison Tool Menu (GOCMENU)** panel, notice that the **Specification Status:** next to option **1** is now **DDL file specified**.

2. Specify the compare target.

   In this scenario, you want Object Comparison Tool to determine the target objects from the Db2 catalog based on the source.

   a) On the **DB2 Object Comparison Tool Menu (GOCMENU)** panel, select option **2 - Specify compare target (old)**.

   b) On the **Specify Compare Target (GOC1)** panel, select option **4 - Target is from the DB2 catalog and objects are automatically selected**.

   c) On the **Specify Target DB2 Location (GOC14)** panel, enter the following information, and press Enter:

   - The location of your subsystem.
   - The name of a data set to use for the target version file. If the data set does not already exist, it is created.

   On the **DB2 Object Comparison Tool Menu (GOCMENU)** panel, notice that the **Specification Status:** next to option **2** is now **Automatic (DB2 catalog extract)**.

   For this scenario, do not specify any masks or ignore fields.

3. Generate the compare job:

   a) On the **DB2 Object Comparison Tool Menu (GOCMENU)** panel, select option **5 - Generate compare job**.

   b) On the **Generate Compare Jobs (GOC5)** panel, specify the following options, and press Enter:

   Worklist name
   TEST

   Suppress DROP of objects
   YES

   PDS for jobs
   TEST

   Prefix for data sets
   TEST

   Single compare job
   YES

   Member name
   COMPARE

   Set all the remaining options to NO, N, or blank.

4. Edit the generated compare job as needed, and submit the job.
5. Check the output to confirm that the job completed successfully.
6. If needed, make any necessary corrections, generate the compare job again, and recheck the comparison report.
7. Apply the changes to the target table:

   a) On the **DB2 Object Comparison Tool Menu (GOCMENU)** panel, select option **5 - Generate compare job**.

   b) On the **Generate Compare Jobs (GOC5)** panel, specify the following additional options, and press Enter:

   Generate apply jobs
   YES

   Generate one job
   YES

   Member prefix
   APPLY

Content of apply job(s)
ALL

Unload method
U

IDENTITY START value
ORIGINAL

Run REORG/REBUILD
A

c) If the **Change Management Prompt (ADB2CMRO)** panel opens, specify NO.
(For this scenario, assume that you are not using Change Management.)

d) On the **Specify Data Set Name for Apply Jobs (GOC5AJ)** panel, enter the name of the data set where you want the apply job generated.

e) Edit the generated compare job as needed, and submit the job.

f) Check the output to confirm that the job completed successfully.

g) Run the generated apply job to change the target table.

Scenario: Copying objects

You can use Db2 Object Comparison Tool to copy objects. For example, you might want to copy objects in your production environment to a test environment.

About this task

For this scenario, assume that you created a new database, PRODDB, in your production environment, and you want to copy the objects in that database to your test environment, in database TESTDB.

Procedure

To copy PRODDB objects to TESTDB, on a different subsystem:

1. Specify the source object definitions to be compared:
   a) On the **DB2 Object Comparison Tool Menu (GOCMENU)** panel, select **Option 1 Specify compare source (new)**, and press Enter.
   b) On the **Specify Compare Source (GOC1)** panel, select **2 - Source is from the DB2 catalog** , and press Enter.
   c) On the **Specify DB2 Source Catalog Extract (GOC12)** panel, specify the following information:
      - In the **Data set name** field, enter the data set name for the version file.

        Tip: Include the date as part of the name. For example: `proddb.v23.D080311`
      - In the **Description** field, enter a description of the source. For example: `production database`

   d) Select **1 - Source is databases from the DB2 catalog**, and press Enter.
   e) On the **Specify Source DB2 Databases (GOC1D)** panel, specify the **I** line command to insert a database to the list.
   f) On the **Compare Add Databases (GOC1DA)** panel, specify a partial data set name and the location name (for example STLEC1) to identify the data set that you want to copy, and press Enter.
   g) On the **Compare Add Databases (GOC1DD)** panel, use the **S** line command to select the database that you want to copy.
   h) Exit back to the **DB2 Object Comparison Tool Menu (GOCMENU)** panel. Notice that the **Specification Status:** next to option **1** is now **Database extract specified**.
2. Specify the target objects:
   a) On the **DB2 Object Comparison Tool Menu (GOCMENU)** panel, select **Option 2 Specify compare target (old)**, and press Enter.

b) On the **Specify Compare Target (GOC1)** panel, select **4 Target is from the DB2 catalog and the objects are automatically selected**, and press Enter.

In this scenario, choose automatic selection, because these objects might already exist in the target.

c) On the **Specify Target DB2 Location (GOC14)** panel, enter the following information, and press Enter:

- The location of your test subsystem.
- The name of a data set to use for the target version file. If the data set does not already exist, it is created.

On the **DB2 Object Comparison Tool Menu (GOCMENU)** panel, notice that the **Specification Status:** next to option **2** is now **Automatic (DB2 catalog extract)**.

3. Specify the masks:

If naming differences exist between objects in the test database and objects in the production database, use masks to account for these naming differences. For example, owner, table name, or table space names might be different.

a) On the **DB2 Object Comparison Tool Menu (GOCMENU)** panel, select **Option 3, Specify Compare Masks**, and press Enter.

b) On the **Specify Compare Masks (GOC3)** panel, complete the following fields, and press Enter:

Mask DSN
The name of the data set for the masks. If the data set does not exist, it is created.

Edit Mask
YES

This scenario shows you how to define masks in a data set. Alternatively, if Change Management is enabled, you can define masks in the Change Management repository.

c) On the **Edit Compare Masks (GOCEDIT)** panel, specify the masks that you want to use.

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)
```

Also consider that you might want to overwrite some values, such as the COMPRESS attribute.

For help in defining masks and overwriting values, see "Translation masks" on page 65 and Mask definitions (IBM Db2 Administration Tool for z/OS 13.1.0).

d) Exit back to the **DB2 Object Comparison Tool Menu (GOCMENU)** panel. Notice that the **Specification Status:** next to option **3** is now **Mask specified**.

4. Specify fields to ignore:

You probably do not want to build test objects exactly the same as production objects. For example, you might want to ignore fields for buffer pools, PRIQTY, or SECQTY.

a) On the **DB2 Object Comparison Tool Menu (GOCMENU)** panel, select **4 - Specify fields to ignore**.

b) On the **Specify Compare Ignores (GOC4)** panel, complete the following fields, and press Enter:

Data Set Name
The name of the data set for the ignores. If the data set does not exist, it is created.

Edit Ignore Fields Specification
YES

This scenario shows you how to define ignores in a data set. Alternatively, if Change Management is enabled, you can define ignores in the Change Management repository.

c) On the **Specify Ignore Fields : Objects (GOCCI)** panel, specify the **U** line command for SYSTABLESPACE.

d) On the **Specify Ignore Fields for object (GOCCIF)** panel, specify the **S** line command for BPOOL, PQTY, and SECQTYI. Press Enter after each selection.

The **Action** column indicates that the field is selected.

e) Press PF3 to return to the **Specify Ignore Fields : Objects (GOCCI)** panel. Notice that the **Ignore Fields** column for SYSTABLESPACE lists `BPOOL, PQTY, SECQTYI`.

f) Exit back to the **DB2 Object Comparison Tool Menu (GOCMENU)** panel. Notice that the **Specification Status:** next to option **4** is now **Ignore fields specified**.

5. Generate compare jobs:

a) On the **DB2 Object Comparison Tool Menu (GOCMENU)** panel, select **5 - Generate compare job**, and press Enter.

b) On the **Generate Compare Jobs (GOC5)** panel, specify the following values, and press Enter:

Worklist name: TEST
Scope Warning Messages: YES
PDF for batch jobs: TEST
Prefix for data sets: TEST
Generate one job: YES
Set all the remaining options to NO or N or the default.

c) If the **Change Management Prompt (ADB2CMRO)** panel opens, specify `NO`.

(For this scenario, assume that you are not using Change Management.)

d) On the **Specify Data Set Name for Apply Jobs (GOC5AJ)** panel, enter the name of a data set where you want the apply jobs generated.

e) Edit the generated JCL job as needed and submit it to run the comparison.

f) Check that the job completed successfully.

6. Check the comparison report.

In the job output, look at the information under the line `OBJECT COMPARISON REPORT`. (For help in evaluating the output, see "Scenario: Comparing a Db2 development catalog to a Db2 production catalog" on page 33.)

7. Correct any problems with the job by changing the masks and ignore fields. Then, regenerate and re-run the comparison job. Repeat this process until the object comparison report contains the differences that you want apply to the target.

8. Regenerate the comparison job and an apply job:

a) On the **DB2 Object Comparison Tool Menu (GOCMENU)** panel, select **5 - Generate compare job**, and press Enter.

b) On the **Generate Compare Jobs (GOC5)** panel, set the **Generate apply jobs** field to `Yes`, and press Enter.

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.

c) If the **Change Management Prompt (ADB2CMRO)** panel opens, specify `NO`.

(For this scenario, assume that you are not using Change Management.)

d) On the **Specify Data Set Name for Apply Jobs (GOC5AJ)** panel, enter the name of the data set where you want the apply job generated.

e) Edit the generated compare job as needed, and submit the job.

f) Check the output to confirm that the job completed successfully.

9. Run the generated apply job to create the new TESTDB objects.

Scenario: Converting partitioned table spaces to partition-by-range universal table spaces

You can use Db2 Object Comparison Tool to change a group of partitioned table spaces to partition-by-range (PBR) universal table spaces (UTS) without having to alter each table space individually.

About this task

This process does not generate batch jobs. However, you can modify existing compare batch jobs to perform a similar function. The key is to create a mask that overwrites the SEGSIZE value of the table spaces.

This scenario assumes that the partitioned table spaces use table-controlled partitioning. If your partitioned table spaces uses index-controlled partitioning, you must convert them to use table controlled-partitioning before following the steps in this scenario. See Converting table spaces to use table-controlled partitioning (Db2 13 for z/OS).

Procedure

To change partitioned table spaces to partition-by-range universal table spaces:

1. Specify the compare source.

   The source of the comparison operation can be a database, which includes all table spaces in that database, or a specified list of table spaces. This scenario specifies the source as a database that contains the partitioned table spaces.

   a) On the **DB2 Object Comparison Tool Menu (GOCMENU)** panel, select option **1 - Specify compare source (new)** and press Enter.

   b) On the **Specify Compare Source (GOC1)** panel, select option **2 - Source is from the DB2 catalog** and press Enter.

   c) On the **Specify DB2 Source Catalog Extract (GOC12)** panel, in the **Data set name** field, specify the name of the data set for the version file for the source.

   d) Select option **1 - Source is databases from the DB2 catalog** and press Enter.

   e) On the **Specify Source DB2 Databases (GOC1D)** panel, use the **I** line command to insert a line in the database list.

   f) On the **Compare Add Databases (GOC1DA)** panel, specify the source database by entering a partial database name and pressing Enter.

   g) On the **Compare Add Databases (GOC1DD)** panel, use **S** line command to select the database that you want to use as the compare source.

2. Specify the compare target.

   In this case, you want the database to be compared to itself, so you need to specify that the target objects are to be automatically selected.

   a) Exit back to the **DB2 Object Comparison Tool Menu (GOCMENU)** panel and select option **2 - Specify compare target (old)**.

   b) On the **Specify Compare Target (GOC1)** panel, select option **4 - Target is from the DB2 catalog and the objects are automatically selected based on the selected source objects** and press Enter.

   c) On the **Specify Target DB2 Location (GOC14)** panel, complete the following fields and press Enter:

   Specify location name:
   The location of the Db2 subsystem.

   Data set name
   The name of the data set for the version file for the target. If the data set does not exist, it is created.

3. Create a mask to overwrite the SEGSIZE value of the table spaces.

a) Exit back to the **DB2 Object Comparison Tool Menu (GOCMENU)** panel and select option **3 - Specify Compare Masks**.

b) On the **Specify Compare Masks (GOC3)** panel, complete the following fields:

Mask DSN
The name of the data set for the masks. If the data set does not exist, it is created.

Edit Mask
YES

c) On the **Edit Compare Masks (GOCEDIT)** panel, insert a line to create a mask to overwrite the SEGSIZE value. For example:

```
SEGSIZE:*,64
```

This line specifies that Db2 Object Comparison Tool is to find all table spaces in the compare scope that match this mask specification (for example, a table space named TS01) and change the value of its SEGSIZE to 64.

4. Generate and run the compare job:

a) Exit back to the **DB2 Object Comparison Tool Menu (GOCMENU)** panel and select option **5 - Generate compare job**.

b) On the **Generate Compare Jobs (GOC5)** panel, specify the appropriate options, including the following settings, and press Enter.

Suppress DROP of objects
YES

Generate apply jobs
NO

Run REORG/REBUILD
A

c) Submit the generated JCL job and check that it runs successfully.

5. Check the object comparison report.

The report shows that the only change to the affected table spaces is the one that was specified by the mask: the ALTER SEGSIZE operation. Additionally, it confirms that the table space changed from partitioned to partition-by-range.

Example message in report:

```
>ADB3320W :SEGSIZE was masked from 0 to 64 for table space DB5772.TS5772. The value might change the
         table space type.
Compare tablespace source(Q79A.Q79A0100) and target(DB5772.TS5772)
    (A)Tablespace change from partitioned to partition-by-range
    (A)Field SEGSIZE changed from 0 to 64
  Tablespace will be altered
```

6. Apply these changes or use Change Management to implement the changes.

As a result, any partitioned table space in the database is now a PBR UTS.

Chapter 4. Comparing Db2 objects

The panel interface in Db2 Object Comparison Tool guides you through the options for comparing Db2 objects. Then, based on those options, the tool generates comparison batch jobs that you can save and reuse.

Before you begin

Before you compare objects, complete the following actions:

- Make sure that the **Get DB2 ZPARM** field (the GETDB2ZP parameter) on the **DB2 Admin Defaults (ADB2P2)** panel is set to YES. For detailed instruction on how to change this value, see Changing defaults (IBM Db2 Administration Tool for z/OS 13.1.0).

  This option enables Object Comparison Tool to get Db2 subsystem parameter values by calling the stored procedure ADMIN_INFO_SYSPARM. These values are needed so that Object Comparison Tool can write a version file for each object that is being compared. *Version files* are snapshots of an object at a particular time and include the object definition.
- If you plan to request that Object Comparison Tool also generate apply jobs, make sure that you are connected to the target Db2 subsystem. *Apply jobs* are jobs that apply changes to the target; these changes are found during the comparison.

  This prerequisite ensures that the apply jobs use the correct libraries. If you are connected to a different Db2 subsystem, you must manually update the apply jobs to use the correct Db2 libraries.

Tip: Before comparing objects, read “Performance considerations for Db2 Object Comparison Tool” on page 48.

Procedure

To compare Db2 objects:

Note: You can specify the W option to be guided directly through the next five steps [options 1 -5 on the **DB2 Object Comparison Tool Menu (GOCMENU)** panel] in succession without returning to the **DB2 Object Comparison Tool Menu (GOCMENU)** panel 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.

1. Specify source objects.
2. Specify target objects.
3. Optional: Specify compare masks.
4. Optional: Specify ignores.
5. Generate a compare batch job.
6. Run the generated batch job to compare the objects.

What to do next

After you run the comparison, check the generated compare report and then optionally apply the changes to the target objects.

Tip: Before you exit Db2 Object Comparison Tool, consider saving your current compare batch job selections for later use. See “Saving dialogs” on page 110.

Related tasks

“Managing and restoring dialogs” on page 111

You can restore, rename, and delete previously saved dialogs.

Related reference

"Object Comparison Tool main menu" on page 32

Use the **DB2 Object Comparison Tool Menu (GOCMENU)** panel to specify the criteria for the comparison that you want to run.

Performance considerations for Db2 Object Comparison Tool

The performance of Db2 Object Comparison Tool can be impacted by several factors.

For optimal performance, avoid the following situations:

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

1. Specifying source objects

The first step in comparing Db2 objects is to specify the definition for the source object or objects. A *source object* is an object as you want it defined. The target object will be compared to this source object and optionally changed to match it.

About this task

The definitions for source objects can be in a DDL file, the Db2 catalog, or a version file. When you specify one of these definition sources, Db2 Object Comparison Tool uses the object definition (and attributes) from that source for the comparison.

Procedure

To specify source object definitions:

1. On the **DB2 Object Comparison Tool Menu (GOCMENU)** panel, specify option 1, and press Enter.
2. On the **Specify Compare Source (GOC1)** panel, specify where you want Object Comparison Tool to retrieve the definitions for the source objects.

```
 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 8. **Specify Compare Source (GOC1)** panel*

You can specify one of the following options:

Option	Description
1	Specifies that the definitions of the source objects are to be retrieved from a *DDL file*, that is, a file that contains SQL CREATE statements.

Option	Description
2	Specifies that the definitions of the source objects are to be extracted from the Db2 catalog. The definitions are extracted for one or more databases, table spaces, or tables and all dependent objects. With option 2, you can optionally specify your own SQL SELECT statement against the catalog to identify the objects.
3	Specifies that the definitions of the source objects are to be retrieved from a previously created version file.
VS	Specifies that the definitions of the source objects are to be extracted from the Db2 catalog, and the objects are selected based on a version scope. Change Management must be enabled to select this option.

3. Optional: If you want to exclude specific objects, specify an exclude specification in the **Exclude Specifications** fields.

   You can specify an existing exclude specification or create a new one. If you specify an existing one, you can edit it.

   For detailed instructions on how to create a new exclude specification or edit an existing one, see "Creating or editing exclude specifications during a comparison" on page 119.
4. Press Enter, and complete one of the following procedures depending on that the option that you chose in step "2" on page 48:

Option	Description
1	"Specifying a DDL file for the source or target definition" on page 49
2	If you want to identify individual objects: "Specifying the Db2 catalog for the source or target definition" on page 51 If you want to specify a SELECT statement against the catalog: "Specifying a SELECT statement for the source or target definition" on page 56
3	"Specifying a version file for the source or target definition" on page 58
VS	"Specifying a version scope for the source or target definition" on page 59

Related concepts

"Terminology in Db2 Object Comparison Tool " on page 12
Db2 Object Comparison Tool uses several terms that are unique to the product.

Specifying a DDL file for the source or target definition

Db2 Object Comparison Tool can use a file that contains data definition language (DDL) for the definitions of the source or target objects. Object Comparison Tool processes everything in the DDL file; objects are not selected based on type or name. If your DDL defines a single table, only that table is used.

Before you begin

This procedure assumes that you have completed the steps in "1. Specifying source objects" on page 48 or "2. Specifying target objects" on page 60 and specified option 1 on the **Specify Compare Source (GOC1)** panel or the **Specify Compare Target (GOC1)** panel. The **Specify Source DDL File (GOC11)** panel or the **Specify Target DDL File (GOC11)** panel should be displayed.

About this task

Note: If the source and target are both DDL, the SYSPRINT data set will list the values for Db2 function level and APPLCOMPAT as NA, because no Db2 connection exists to obtain accurate values.

Procedure

To specify a DDL file for the source or target definition:

1. On the **Specify Source DDL File (GOC11)** panel or the **Specify Target DDL File (GOC11)** panel, in the **DDL data set** field, specify the name of the data set that contains the DDL for the source or target object or objects:

```
GOC11 e                    Specify Source 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                       |
 ------------------------------------------------------------------------------
```

   Note: If Change Management (CM) is not enabled, the **Owner** and **Name** fields are not available on this panel.

   *Figure 9. **Specify Source DDL File (GOC11)** panel*

   The data set that you specify must contain valid SQL statements and must adhere to TSO naming conventions. The data set can be either of the following 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.

2. Specify where the version file will be written.

   The *version file* will contain the object definitions that are extracted by the Object Comparison Tool DDL extract program and become input to the compare process. You can specify one of the following locations for the version file:

   Output data set
   To specify that the version file be written to an output data set, enter a value in the **Data set name** field. 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 16 KB (`RECFM=Vx,LRECL=16384`)

   If the specified data set does not exist, it is created. If an existing data set is specified, it is overwritten.

CM database
To specify that the version file be written to the CM database, specify values in the version table **Owner** and **Name** fields. To select a version file from the CM database, specify a question mark (?) in the field. You can choose a value on the subsequent **CM Versions (ADB2C41)** panel.

If the **Owner** and **Name** fields are not visible, CM is not enabled, and you must specify an output data set instead.

If you select the option to store the version file in the CM 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.

3. Optional: In the **Description** field, specify a description of the source or target Db2 objects.

   The description is printed in the comparison report, placed in the header record of the version file, and used to describe the common properties of the Db2 objects.

4. Press Enter.

   The **DB2 Object Comparison Tool Menu (GOCMENU)** panel is displayed. `DDL file specified` is listed as the **Specification Status** for the source or target, depending on which one you specified.

What to do next

If you specified the source object definitions, specify target object definitions.

If you specified the target object definitions, specify compare masks (optional), specify ignore fields (optional), or generate a compare batch job.

Related concepts

“Batch DDL file extraction program ” on page 137
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 information

Change Management (CM) (IBM Db2 Administration Tool for z/OS 13.1.0)

Specifying the Db2 catalog for the source or target definition

Db2 Object Comparison Tool can extract the source or target object definitions from the Db2 catalog.

Before you begin

This procedure assumes that you have completed the steps in “1. Specifying source objects” on page 48 or “2. Specifying target objects” on page 60 and specified option 2 on the **Specify Compare Source (GOC1)** panel or the **Specify Compare Target (GOC1)** panel. The **Specify DB2 Source Catalog Extract (GOC12)** panel or the **Specify DB2 Target Catalog Extract (GOC12)** panel should be displayed.

Procedure

To specify the Db2 catalog for the source or target definition:

1. On the **Specify DB2 Source Catalog Extract (GOC12)** panel or the **Specify DB2 Target Catalog Extract (GOC12)** panel, specify where the version file is to be written:

```
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
   5 - Target is the result of an SQL SELECT statement

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 10. ***Specify DB2 Source Catalog Extract (GOC12)*** *panel*

The *version file* will contain the object definitions that are extracted by the Object Comparison Tool DDL extract program and become input to the compare process. You can specify one of the following locations for the version file:

Output data set
To specify that the version file be written to an output data set, enter a value in the **Data set name** field. 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 16 KB (`RECFM=Vx,LRECL=16384`)

If the specified data set does not exist, it is created. If an existing data set is specified, it is overwritten.

CM database
To specify that the version file be written to the CM database, specify values in the version table **Owner** and **Name** fields. To select a version file from the CM database, specify a question mark (?) in the field. You can choose a value on the subsequent **CM Versions (ADB2C41)** panel.

If the **Owner** and **Name** fields are not visible, CM is not enabled, and you must specify an output data set instead.

If you select the option to store the version file in the CM 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.

2. Optional: In the **Description** field, specify a description of the source or target Db2 objects.

   The description is printed in the comparison report, placed in the header record of the version file, and used to describe the common properties of the Db2 objects.

3. Specify which object definitions you want extracted from the Db2 catalog for the source or target by specifying one of the following options, and press Enter:

Option	Description
1	Databases and all dependent objects, such as table spaces, tables, views, indexes, aliases, synonyms, and so forth. You can subsequently select which databases. **Tip:** If the database scope is too large, you can select table spaces or tables.
2	Table spaces and all dependent objects. You can subsequently select which table spaces.
3	Tables and all dependent objects. You can subsequently select which tables.

Option	Description
4	Schema-based objects, such as user-defined functions, user-defined types, stored procedures, and sequences. 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, you do not need to define a trigger as a separate object. You can choose this option (4) separately or as an additional option combined with option 1, 2, or 3.

If you want to specify a SELECT statement (option **5**), see "Specifying a SELECT statement for the source or target definition" on page 56.

4. On the one of the following **Specify Source** or **Specify Target** panels, issue the I line command to add objects to the list, and press Enter:
   - **Specify Source DB2 Databases (GOC1D)** panel
   - **Specify Source DB2 Table Spaces (GOC1S)** panel
   - **Specify Source DB2 Tables (GOC1T)** panel
   - **Specify Source DB2 Schema (GOC1C)** panel
   - **Specify Target DB2 Databases (GOC1D)** panel
   - **Specify Target DB2 Table Spaces (GOC1S)** panel
   - **Specify Target DB2 Tables (GOC1T)** panel
   - **Specify Target DB2 Schema (GOC1C)** panel

   For example:

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

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

 Select Database  Location
         *         *
------ --------  ----------------
I      ........  ................
       AB        STPLEX4A_DSN8
       ADBDB010  STPLEX4A_DSN8
       ADBDCHD   STPLEX4A_DSN8
```

*Figure 11. **Specify Source DB2 Databases (GOC1D)** panel*

Note: The list of objects is empty the first time that the **Specify Source** or **Specify Target** panel is displayed.

Tip: If objects are listed that you do not want to include, use the D line command to remove them from the list. To remove all objects from the list, issue the RESET command.

5. On one of the following **Compare Add** panels, specify the Db2 catalog search criteria for the objects, and press Enter:
   - **Compare Add Databases (GOC1DA)** panel
   - **Compare Add Table Spaces (GOC1SA)** panel
   - **Compare Add Tables (GOC1TA)** panel
   - **Compare Add Schema (GOC1CA)** panel

For example:

```
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 12. **Compare Add Databases (GOC1DA)** panel*

Depending on which panel is displayed, specify one or more of the following values:

Partial database name
The database name.

Partial table space name
The table space name

Partial table owner
The authorization ID of a table owner

Partial table name
The table name

Partial schema name
The schema name

Location name
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.

One of the following **Compare Add** panels displays the qualifying objects:

- **Compare Add Databases (GOC1DD)** panel
- **Compare Add Table Spaces (GOC1SD)** panel
- **Compare Add Tables (GOC1TD)** panel
- **Compare Add Schema (GOC1CD)** panel

For example:

```
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
       *        *
------ -------- ------
       DSG24D0G
       DSG24D0X
       DSG24D1Z
       DSNAE71A
       DSNAE71P
       DSNATPDB
       DSNDB04
       DSNDB06
       DSNDB07
       DSNDPSM
       DSNRGFDB
       DSNRLST
       DSNRTSDB
       DSN8CDDB
       DSN8D71L
       DSN8TEMP
```

*Figure 13. **Compare Add Databases (GOC1DD)** panel*

6. Issue the S line command next to the objects that you want to select, and press Enter.

   The panel shows which objects are added. For example, the following panel shows that databases DSG24D0G and DSG24D1Z were added to the source for the comparison:

```
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
       DSNDPSM
       DSNRGFDB
       DSNRLST
       DSNRTSDB
       DSN8CDDB
       DSN8D71L
       DSN8TEMP
```

*Figure 14. **Compare Add Databases (GOC1DD)** panel*

   Tip: If you want to select all objects listed, issue the ALL command, which automatically selects the objects and exits the panel.

7. Exit (PF3) back to one of the following **Specify Source** or **Specify Target** panels:
   - **Specify Source DB2 Databases (GOC1D)** panel
   - **Specify Source DB2 Table Spaces (GOC1S)** panel
   - **Specify Source DB2 Tables (GOC1T)** panel
   - **Specify Source DB2 Schema (GOC1C)** panel

- **Specify Target DB2 Databases (GOC1D)** panel
- **Specify Target DB2 Table Spaces (GOC1S)** panel
- **Specify Target DB2 Tables (GOC1T)** panel
- **Specify Target DB2 Schema (GOC1C)** panel

The objects that you added are listed.

Restriction: You cannot compare objects from different locations, so ensure that all listed objects are from the same location.

8. Exit (PF3) to the **DB2 Object Comparison Tool Menu (GOCMENU)** panel.

   object types `extract specified` is listed as the **Specification Status** for the source or target, depending on which one you specified.

What to do next

If you specified the source object definitions, specify target object definitions.

If you specified the target object definitions, specify compare masks (optional), specify ignore fields (optional), or generate a compare batch job.

Specifying a SELECT statement for the source or target definition

You can specify a SELECT statement against the Db2 catalog to identify the source or target for a comparison. Db2 Object Comparison Tool uses all of the objects that are returned by the query for the source or target definition. The advantage of using a SELECT statement to identify the source or target is that you can use clauses to filter the object list.

Before you begin

This procedure assumes that you have completed the steps in "1. Specifying source objects" on page 48 or "2. Specifying target objects" on page 60 and specified option 2 on the **Specify Compare Source (GOC1)** panel or the **Specify Compare Target (GOC1)** panel. The **Specify DB2 Source Catalog Extract (GOC12)** panel or the **Specify DB2 Target Catalog Extract (GOC12)** panel should be displayed.

Procedure

To specify a SELECT statement for the source or target definition:

1. On the **Specify DB2 Source Catalog Extract (GOC12)** panel or the **Specify DB2 Target Catalog Extract (GOC12)** panel, specify where the version file is to be written:

```
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
   5 - Source is the result of an SQL SELECT statement

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 15. **Specify DB2 Source Catalog Extract (GOC12)** panel*

The *version file* will contain the object definitions that are extracted by the Object Comparison Tool DDL extract program and become input to the compare process. You can specify one of the following locations for the version file:

Output data set
To specify that the version file be written to an output data set, enter a value in the **Data set name** field. 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 16 KB (RECFM=Vx,LRECL=16384)

If the specified data set does not exist, it is created. If an existing data set is specified, it is overwritten.

CM database
To specify that the version file be written to the CM database, specify values in the version table **Owner** and **Name** fields. To select a version file from the CM database, specify a question mark (?) in the field. You can choose a value on the subsequent **CM Versions (ADB2C41)** panel.

If the **Owner** and **Name** fields are not visible, CM is not enabled, and you must specify an output data set instead.

If you select the option to store the version file in the CM 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.

2. Optional: In the **Description** field, specify a description of the source or target Db2 objects.

   The description is printed in the comparison report, placed in the header record of the version file, and used to describe the common properties of the Db2 objects.

3. Specify option 5, and press Enter.

   The **Define SQL SELECT Statement For SOURCE (GOCSQ)** panelor **Define SQL SELECT Statement For TARGET (GOCSQ)** panel is displayed:

```
GOCSQ min ----------- Define SQL SELECT Statement For SOURCE ------------ 12:12
Option ===>



1 - Edit SQL SELECT statement

Data set name . .
                  (Default if blank: 'TS6462.GOCSQL.SOURCE')

Location name . . RS22DC1A          >
                  (Default if blank: RS22DD1A)
```

*Figure 16. **Specify Source DB2 Databases (GOC1D)** panel*

4. Optional: Specify a data set name and the location of the subsystem that contains the objects to be compared.

   If you do not specify a data set name, a temporary data set will be created with the default name listed. If you do not specify a location, the local subsystem will be used.

   If the specified data does not already exist, it will be created.

5. Specify option 1, and press Enter.
6. Write or edit the SQL statement as needed.

   The SELECT statement must query the Db2 catalog and return the following columns:

   - TYPE CHAR(2)
   - QUAL VARCHAR(128)
   - NAME VARCHAR(128)

The SELECT statement can also optionally return the following additional columns:

- VERSION VARCHAR(122)
- INCLUDE CHAR(8)
- XDTYPE CHAR(8) (exclude)

For example, the following query returns table spaces created by TS3071:

```
SELECT 'TS' AS TYPE, DBNAME AS QUAL, NAME
  FROM SYSIBM.SYSTABLESPACE
  WHERE CREATOR = 'TS3071'
```

Note: If the SQL returns XDTYPE requests, make sure that the SQL returns those rows first.

7. Exit (PF3) to the **DB2 Object Comparison Tool Menu (GOCMENU)** panel.

   `SQL SELECT (DB2 catalog extract)` is listed as the **Specification Status** for the source or target, depending on which one you specified.

What to do next

If you specified the source object definitions, specify target object definitions.

If you specified the target object definitions, specify compare masks (optional), specify ignore fields (optional), or generate a compare batch job.

Specifying a version file for the source or target definition

If the object or objects were previously part of another comparison, you can specify an existing version file to be used for the source or target definition. A *version file* is created by Db2 Object Comparison Tool during the compare process. This file includes the object definitions at a particular point in time.

Before you begin

This procedure assumes that you have completed the steps in "1. Specifying source objects" on page 48 or "2. Specifying target objects" on page 60 and specified option 3 on the **Specify Compare Source (GOC1)** panel or the **Specify Compare Target (GOC1)** panel. The **Specify Source Compare Version File (GOC13)** panel or the **Specify Target Compare Version File (GOC13)** panel should be displayed.

Tip: if the version file is old, consider converting to the latest Db2 version to save time. See Chapter 7, "Converting version files to the latest Db2 version," on page 129.

Procedure

On the **Specify Source Compare Version File (GOC13)** panel or the **Specify Target Compare Version File (GOC13)** panel, specify the version file that you want to use, and press Enter.

```
Compare ------------- Specify Source Compare Version File ---------------------
Command ===>



Specify compare version file (Source):
  Version table entry:
    Owner . . .           >             (? to look up)
    Name  . . .                      >  (? to look up)

  Data set:
    Data set name . . .
    Data set rewritable . . .           (Yes/No)
    Estimated record num. . . 0         (Only for DS cannot be rewritten)
```

*Figure 17. **Specify Source Compare Version File (GOC13)** panel*

You can specify one of the following sources for the previously created version file that you want to use as the source or target:

CM database
To specify that the version file is in the CM database, specify values in the version table **Owner** and **Name** fields. To select a version file from the CM database, specify a question mark (?) in the field. You can choose a value on the subsequent **CM Versions (ADB2C41)** panel.

If the **Owner** and **Name** fields are not visible, CM is not enabled, and you must specify a data set instead.

Data set
To specify that the version file is in a data set, enter a value in the **Data set name** field.

If you specify a data set, also specify whether the version file can be rewritten (in the **Data set rewritable** field). The default value is YES. If you specify NO, also specify a value for the **Estimated record num** field. If the version file can be rewritten, Db2 Object Comparison Tool counts this record number for you and writes it in the version file.

Results
The **DB2 Object Comparison Tool Menu (GOCMENU)** panel is displayed, and `Compare version file specified` is listed as the **Specification Status** for the source or target, depending on which one you specified.

What to do next

If you specified the source object definitions, specify target object definitions.

If you specified the target object definitions, specify compare masks (optional), specify ignore fields (optional), or generate a compare batch job.

Specifying a version scope for the source or target definition

If Change Management (CM) is enabled, you can select a version scope for your source or target. A *version scope* is a predefined set of objects. Version scope definitions are stored in CM tables and can include databases, table spaces, tables, indexes, views, stored procedures, triggers, and other objects. If you specify a version scope as your source or target, the definitions of the specified objects are retrieved from the Db2 catalog for the comparison.

Before you begin
This procedure assumes that you have completed the steps in "1. Specifying source objects" on page 48 or "2. Specifying target objects" on page 60 and specified option VS on the **Specify Compare Source (GOC1)** panel or the **Specify Compare Target (GOC1)** panel. The **Specify Source Version Scope (GOC1VS)** panel or the **Specify Target Version Scope (GOC1VS)** panel should be displayed.

Procedure

To specify a version scope for the source or target definition:

1. On the **Specify Source Version Scope (GOC1VS)** panel or the **Specify Target Version Scope (GOC1VS)** panel, specify values in the version table **Owner** and **Name** fields to identify the version scope that you want to use to extract definitions from the Db2 catalog:

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

Specify version scope (Source):
   Owner . . . . .            >           (? to look up)
   Name  . . . . .                     >  (? to look up)

 Specify compare version file output:
   Data set name .

 Enter a description (optional):
   Description . .
```

*Figure 18. **Specify Source Version Scope (GOC1VS)** panel*

2. In the **Data set name** field, specify the output data set name for the version file.

   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=6144`)

   If the specified data set does not exist, it is created. If an existing data set is specified, it is overwritten.
3. Optional: In the **Description** field, specify a description of the source or target Db2 objects.

   The description is placed in the header record of the version file.
4. Press Enter.

   The **DB2 Object Comparison Tool Menu (GOCMENU)** panel is displayed, and `Version scope specified (DB2 catalog)` is listed as the **Specification Status** for the source or target, depending on which one you specified.

What to do next

If you specified the source object definitions, specify target object definitions.

If you specified the target object definitions, specify compare masks (optional), specify ignore fields (optional), or generate a compare batch job.

Related information

Version scopes (IBM Db2 Administration Tool for z/OS 13.1.0)

2. Specifying target objects

After you specify the comparison source, the next step is to specify the target. The *target* is the object or objects that you want to compare to the source.

Procedure

To specify the target object definitions:

1. On the **DB2 Object Comparison Tool Menu (GOCMENU)** panel, specify option 2, and press Enter.
2. On the **Specify Compare Target (GOC1)** panel, specify where you want Object Comparison Tool to retrieve the definitions for the target objects, and press Enter:

```
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)
```

You can specify one of the following options:

Option	Description
1	Specifies that the definitions of the target objects are to be retrieved from a *DDL file*, that is, a file that contains SQL CREATE statements.
2	Specifies that the definitions of the target objects are to be extracted from the Db2 catalog. The definitions are extracted for one or more databases, table spaces, or tables and all dependent objects. With option 2, you can optionally specify your own SQL SELECT statement against the catalog to identify the objects.
3	Specifies that the definitions of the target objects are to be retrieved from a previously created version file.
4	Specifies that the target objects are selected based on the source objects.
VS	Specifies that the definitions of the targets objects are to be extracted from the Db2 catalog, and the objects are selected based on a version scope. Change Management must be enabled to select this option.

3. Optional: If you want to exclude specific objects, specify an exclude specification in the **Exclude Specifications** fields.

   You can specify an existing exclude specification or create a new one. If you specify an existing one, you can edit it.

   For detailed instructions on how to create a new exclude specification or edit an existing one, see "Creating or editing exclude specifications during a comparison" on page 119.

4. Press Enter, and complete one of the following procedures depending on that the option that you chose in step "2" on page 60:

Option	Description
1	"Specifying a DDL file for the source or target definition" on page 49
2	If you want to identify individual objects: "Specifying the Db2 catalog for the source or target definition" on page 51 If you want to specify a SELECT statement against the catalog: "Specifying a SELECT statement for the source or target definition" on page 56
3	"Specifying a version file for the source or target definition" on page 58
4	"Specifying that the target definition is automatically selected based on the source" on page 62
VS	"Specifying a version scope for the source or target definition" on page 59

Related tasks

"1. Specifying source objects" on page 48
The first step in comparing Db2 objects is to specify the definition for the source object or objects. A *source object* is an object as you want it defined. The target object will be compared to this source object and optionally changed to match it.

Specifying that the target definition is automatically selected based on the source

The target can be defined based on the source objects. In this case, Object Comparison Tool uses the source object names, in combination with masks and renames, to determine the target object names. Object Comparison Tool then extracts the definitions of the target objects from the Db2 catalog accordingly.

Before you begin

This procedure assumes that you have completed the steps in "2. Specifying target objects" on page 60 and specified option 4 on the **Specify Compare Target (GOC1)** panel. The **Specify Target DB2 Location (GOC14)** panel should be displayed.

Important: If you specify option 4 and the source is not a table space, objects that exist only in the target might be dropped. You can request that a warning message be issued for this situation by using the **Scope Warning Messages** option on the **Generate Compare Jobs (GOC5)** panel or the CM batch parameter SCOPE_WARNING.

Procedure

To specify that the target definition is automatically selected based on the source:

1. On the **Specify Target DB2 Location (GOC14)** panel, specify the Db2 location name for the target and the name of the output data set for the version file:

```
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                      |
--------------------------------------------------------------------------------
```

   The specified output 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 16 KB (RECFM=Vx,LRECL=16384)

   If the specified data set does not exist, it is created. If an existing data set is specified, it is overwritten.
2. Optional: In the **Description** field, specify a description of the Db2 objects that you are comparing.
   The description is included in the comparison report and in the header record of the version file.
3. Press Enter.

Object Comparison Tool determines the set of objects that are defined in the source version file and extracts the definitions of these objects from the Db2 catalog for the target.

The **DB2 Object Comparison Tool Menu (GOCMENU)** panel is displayed, and the **Specification Status** for the target is listed as `Automatic (DB2 catalog extract)`.

What to do next

Specify compare masks (optional), specify ignore fields (optional), or generate a compare batch job.

Related information

Version scopes (IBM Db2 Administration Tool for z/OS 13.1.0)

3. Specifying compare masks

If you need to account for different naming conventions between the objects you are comparing or overwrite certain attributes, specify masks. This step is optional.

You can define a mask either in a data set or in a table in the Change Management repository. Storing masks in a data set makes copying mask files easy. Storing masks in a table makes them easy to share, manage, and recover.

Tip: If you want to mask the schema and the owner, you must specify masks for both schema and owner, even if the values are the same.

Before you begin

If you plan to define your masks in a table, Change Management must be enabled and the repository tables must be defined as part of the customization process of Db2 Administration Tool.

Procedure

To specify compare masks:

1. On the **DB2 Object Comparison Tool Menu (GOCMENU)** panel, specify option 3.
2. If you want to use a table in the Change Management repository for your masks, complete the following steps. Otherwise, if you want to use a data set for your masks, skip to step "3" on page 64.
   a) On the **Specify Compare Masks (GOC3)** panel, specify an owner and name. This name identifies a row in the mask table in the Change Management database. That row contains (or will contain) the masks that you want to use for the comparison operation. You can specify either an existing name to identify an existing row in the mask table or a specify a new name to create a new row in the mask table.

      Tip: If the **Owner** and **Name** fields are not displayed, Change Management is not enabled. Either enable it or use a data set for your masks (as described in step "3" on page 64).

```
GOC3 re  ---------------------- Specify Compare Masks --------------------------
Option ===>


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

*Figure 19. **Specify Compare Masks (GOC3)** panel*

   b) In the **Edit Mask** field, specify whether you want to edit this set of masks, and press Enter. If the values that you specified in the **Owner** and **Name** fields do not identify an existing set of masks, you must specify YES.

      One of the following panels is displayed:

- If you specified the name of an existing set of masks and NO for **Edit Mask**, the **DB2 Object Comparison Tool Menu (GOCMENU)** panel is displayed. Notice that for step **3 - Specify compare masks**, the **Specification Status** field shows `Mask specified`. You have completed specifying your compare masks. You can skip the rest of this procedure and continue the process of comparing Db2 objects.
- If you specified the name of an existing set of masks and YES for **Edit Mask**, the **Mask Lines (ADB2C2L)** panel is displayed. Skip to step "2.f" on page 64.
- If you specified a new name for a set of masks, the **Insert Mask (ADB2C22)** panel is displayed:

```
 ADB2C22 n ---------------- CM - Insert Mask ---------------------------- 12:25
Command ===>


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

*Figure 20. **Insert Mask (ADB2C22)** panel*

c) Optional: In the **comment** field, enter a description for this set of masks.

d) Press Enter to insert a row into the Change Management repository table for this new set of masks.

e) Press F3 to go to the **Mask Lines (ADB2C2L)** panel where you can define the masks.

f) Insert and update lines as needed to define the masks that you want to use for the comparison.

For example, on the following panel, the TBNAME mask specifies that any table names of TB_TEST are to be translated to TB_PROD for the comparison. (This mask also affects the child masks SYNNAME, ALNAME, and VWNAME.)

For a complete list of mask names and syntax, see Mask definitions (IBM Db2 Administration Tool for z/OS 13.1.0).

```
ADB2C2L n ----------------- CM - Mask Lines ------------------ Row 1 to 1 of 1
Command ===>                                                Scroll ===> PAGE

Mask lines for mask "MYID"."MYMASK"
Commands: CANCEL
Line commands:
  I - Insert  D - Delete  R - Repeat  M - Move  A - After  B - Before

Sel Sequence Req 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 21. **Mask Lines (ADB2C2L)** panel*

g) Exit back to the **DB2 Object Comparison Tool Menu (GOCMENU)** panel. Notice that for step **3 - Specify compare masks**, the **Specification Status** field shows `Mask specified`. You have completed specifying your compare masks. You can skip the rest of this procedure and continue the process of comparing Db2 objects.

3. If you want to use a data set for your masks, complete the following steps:

a) On the **Specify Compare Masks (GOC3)** panel, in the **Mask DSN** field, specify the name of the data set.

You can specify an existing data set that already contains masks or a new data set that you want to use for masks. If the specified data set exists, it is reused. Otherwise, it is created.

The mask data set must adhere to TSO naming conventions and be one of the following types:

- A fixed-block sequential data set (RECFM=F*x*)
- A member of a partitioned data set with a record length of 80 (LRECL=80)

b) In the **Edit Mask** field, specify whether you want to edit the mask data set by using ISPF edit, and press Enter. If the data set is new or does not contain mask definitions, specify YES.

One of the following panels is displayed:

- If you specified NO for **Edit Mask**, the **DB2 Object Comparison Tool Menu (GOCMENU)** panel is displayed. Notice that for step **3 - Specify compare masks**, the **Specification Status** field shows `Mask specified`. You have completed specifying your compare masks. You can skip the rest of this procedure and continue the process of comparing Db2 objects.
- If you specified YES for **Edit Mask**, the **Edit Compare Masks (GOCEDIT)** panel is displayed.

c) On the **Edit Compare Masks (GOCEDIT)** panel, add and change mask definitions as needed, and issue the SAVE command.

For a listing of content that is displayed on the **Edit Compare Masks (GOCEDIT)** panel, see "Mask data set" on page 68.

For a complete list of mask names and syntax, see Mask definitions (IBM Db2 Administration Tool for z/OS 13.1.0).

d) When you are done making changes, exit back to the **DB2 Object Comparison Tool Menu (GOCMENU)** panel. Notice that for step **3 - Specify compare masks**, the **Specification Status** field shows `Mask specified`.

What to do next

Optionally specify ignore fields or generate a compare batch job.

Related concepts

"Translation masks" on page 65
In Object Comparison Tool, you can use translation masks to account for differences in naming conventions between source and target objects when doing a comparison. You can also use masks to overwrite values for object attributes.

Related information

Masks (IBM Db2 Administration Tool for z/OS 13.1.0)

Translation masks

In Object Comparison Tool, you can use translation masks to account for differences in naming conventions between source and target objects when doing a comparison. You can also use masks to overwrite values for object attributes.

This purpose is slightly different than the way masks are used in Db2 Administration Tool. In Db2 Admin Tool, you use masks to change the naming conventions that are used in the generated SQL. In Object Comparison Tool, masks are mainly used for translation. A name in the source can be translated with the mask so that it matches a name in the target. For example, if the source database is named SSEMMDB1 and the target is SSEMMDBA, a mask can tell Object Comparison Tool to compare the two databases even though they have different names.

For mask syntax, see Mask definitions (IBM Db2 Administration Tool for z/OS 13.1.0). The syntax is different depending on whether you are specifying a mask to translate names or a mask to overwrite attribute values.

If you specify both a translation mask and ignore fields, the ignore fields specification overrides the mask.

Masks that translate names

Object Comparison Tool supports three types of translation masks to process names:

AUTHID masks
AUTHID masks are applied to all fields that contain Db2 authorization IDs, such as OWNER and CREATOR.

For example, the following mask specifies that all authorization IDs that have the value SYSIBM in the source are translated to COPY:

```
AUTHID: SYSIBM, COPY
```

With the following example mask, an owner of PROD01 in the source is translated to PRODDB01.

```
AUTHID: *01*, *DB01*
```

You can also specify that you want to translate names for only specific authorization IDs, such as the table space owner. In that case, specify TSOWNER instead of AUTHID. For a complete list of AUTHID masks, see Mask definitions (IBM Db2 Administration Tool for z/OS 13.1.0).

NAME masks
NAME masks are applied to all fields that name objects.

For example, the following mask specifies that any name that starts with ABC in the source is translated to a name that starts with DEF.

```
NAME: ABC*,DEF*
```

With the following example mask, name HLQ47D9 in the source is translated to NEW479 before it is compared with the target.

```
NAME: HLQ*D*, NEW**
```

You can also specify that you want to translate names for only specific types of objects, such as table spaces. In that case, specify TSNAME instead of NAME. For a complete list of name masks, see Mask definitions (IBM Db2 Administration Tool for z/OS 13.1.0).

RENAME specifications
RENAME specifications indicate that an object in the source was renamed and should be related to an existing object in the target.

The syntax for RENAME is:

```
renameobj:old-name,new-name
```

where *old-name* is the previous name of the object and *new-name* is the new name of the object. You can use the wildcard character, an asterisk (*), in the object names.

renameobj is a keyword that indicates the object that was renamed. *renameobj* can have one of the following values:

RENAMEDB
A database was renamed.

Example syntax: RENAMEDB:X*,P*

RENAMETS
A table space was renamed.

Example syntax: RENAMETS:X*.X*,P*.P*

RENAMETB
A table was renamed.

Example syntax: RENAMETB:*.X*,*.P*

RENAMEIX
An index was renamed.

Example syntax: `RENAMEIX:*.X*,*.P*`

RENAMEGV
A global variable was renamed.

Example syntax: `RENAMEGV:*.GVT*,*.GVS*`

RENAMECOL
A column was renamed.

Example syntax: `RENAMECOL:OWNERA.MYTB.OLDCOLNAME,NEWCOLNAME`

An error message is generated in any of the following situations:

- 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 also specify masks to translate names for a specific object, such a specific table space instead of all table spaces. For details, see Mask definitions (IBM Db2 Administration Tool for z/OS 13.1.0).

When you specify compare masks, you do not have to specify the same number of characters for both the input mask and the output mask.

Masks that overwrite attributes

The following example mask specifies that table spaces that start with TESTHRTS in the TESTDB database are to be compressed.

```
COMPRESS: TESTDB.TESTHRTS*, YES
```

The following example mask specifies that the SEGSIZE attribute for all source table spaces is to be changed to 64.

```
SEGSIZE:*,64
```

Using masks to overwrite attributes can be useful when you want to overwrite attributes for a large group of objects. For an example, see “Scenario: Converting partitioned table spaces to partition-by-range universal table spaces” on page 45

For more flexibility, you can also use a REXX user exit to specify the overwrite value for table space and index space attributes. For more information about these REXX user exits, see Specifying a REXX user exit for the overwrite value (IBM Db2 Administration Tool for z/OS 13.1.0).

Db2 catalog records and associated masks

For a list of the Db2 catalog columns that correspond to each mask, see Db2 catalog columns and the corresponding masks (IBM Db2 Administration Tool for z/OS 13.1.0).

Mask processing

Masks are applied to the source objects before they are compared with the target objects. Before the comparison process, any masks are applied to Db2 catalog fields in the version file for the source object so that the names match the naming convention of the target object.

You can use one or more translation masks on the source object to make it match the target object. Masks that translate names are processed first and then any masks that specify overwrite values are applied. Within each of those categories, masks are processed in the order that you list them.

When a value is translated, such as a name, the masks are processed one by one until a match is detected. A match means that both of the following conditions are true:

- The mask name is applicable to the value. For example, for a table name, mask names TBNAME and NAME are applicable.
- The value conforms to *inputmask* in the mask syntax. For example, PRODTAB1 conforms to mask PROD*1.

The value is translated based on the *outputmask* value in the syntax, 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.

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.

Related concepts

"Special considerations for comparing DB2 objects" on page 146
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

"3. Specifying compare masks" on page 63

Mask data set

The mask data set is prepopulated with information about the available masks.

When you edit a mask data set, the **Edit Compare Masks (GOCEDIT)** panel displays the content of this data set. The following screen shows this panel as displayed when it first opens. The message lines (identified by ==MSG) list the available translation mask names. These lines also show the hierarchy of the mask names.

```
****** ***************************** Top of Data ******************************
==MSG>
==MSG>  Mask Syntax:
==MSG>    field:[qual<.name>:]inmask,outmask
==MSG>  Fields (hierarchy):
==MSG>    SINGLECH
==MSG>    COLNAME
==MSG>    NAME
==MSG>      DBNAME,TSNAME,IXNAME,UDFNAME,CONSNAME,
==MSG>      UDTNAME,COLLNAME,PKGNAME,PGMNAME,PLNNAME
==MSG>      DBRMNAME,STPNAME,SFNAME,TGNAME,GRPNAME,
==MSG>      VCATNAME,GBPNAME,TCNAME,PMNAME,MKNAME
==MSG>      SEQNAME,GVNAME
==MSG>      TBNAME
==MSG>        SYNNAME,ALNAME,VWNAME
==MSG>      BPNAME
==MSG>        TSBPNAME,IXBPNAME
==MSG>      SGNAME
==MSG>        TSSGNAME,IXSGNAME
==MSG>    AUTHID
==MSG>      SQLID
==MSG>      SCHEMA
==MSG>        IXSCHEMA,PMSCHEMA,MKSCHEMA,SETPATHSC
==MSG>        TGSCHEMA,UDTSCHEMA,SEQSCHEMA,STPSCHEMA
==MSG>        UDFSCHEMA,GVSCHEMA
==MSG>        TBSCHEMA
==MSG>          ALSCHEMA,VWSCHEMA,SYNSCHEMA
==MSG>      OWNER
==MSG>        DBOWNER,TSOWNER,IXOWNER,SGOWNER
==MSG>        PKGOWNER
==MSG>        TBOWNER
==MSG>      GRANTID
==MSG>        GRANTOR,GRANTEE
==MSG>      ROLE
==MSG>        DBROLE,TSROLE,TBROLE,IXROLE
==MSG>    XMLSCHID
==MSG>    WLMENV
==MSG>    LOCATION
==MSG>
```

*Figure 22. **Edit Compare Masks (GOCEDIT)** panel, part 1*

```
==MSG> Overwrite Syntax:
==MSG>    Field:inmask,Overwrite_value
==MSG>  Fields:              Overwrite values:
==MSG>    COMPRESS           YES,NO,REXX exit (table spaces and indexes)
==MSG>      TSCOMPRES        YES,NO,FIXED,HUFFMAN,REXX exit (table spaces only)
==MSG>      IXCOMPRES        YES,NO,REXX exit (indexes only)
==MSG>    SEGSIZE            n (4-64 must be multiple of 4),REXX exit
==MSG>      TSDSSIZE         nG,REXX exit (table spaces only)
==MSG>      IXDSSIZE         nG,REXX exit (indexes only)
==MSG>    PRIQTY             n,n%,REXX exit (table spaces and indexes)
==MSG>      TSPRIQTY         n,n%,REXX exit (table spaces only)
==MSG>      IXPRIQTY         n,n%,REXX exit (indexes only)
==MSG>    SECQTY             n,n%,REXX exit (table spaces and indexes)
==MSG>      TSSECQTY         n,n%,REXX exit (table spaces only)
==MSG>      IXSECQTY         n,n%,REXX exit (indexes only)
==MSG>    DEFER              YES,NO,REXX exit (indexes only)
==MSG>    DEFINE             YES,NO,REXX exit (table spaces and indexes)
==MSG>      TSDEFINE         YES,NO,REXX exit (table spaces only)
==MSG>      IXDEFINE         YES,NO,REXX exit (indexes only)
==MSG>    HASHSPC            nK,nM,nG,REXX exit
==MSG>    TBINLOBL           n,REXX exit (tables only)
==MSG>    DTINLOBL           n,REXX exit (distinct types only)
==MSG>    AUDIT              CHANGES,ALL,NONE,REXX exit (tables only)
==MSG>    CLOSE              YES,NO,REXX exit (table spaces and indexes)
==MSG>      TSCLOSE          YES,NO,REXX exit (table spaces only)
==MSG>      IXCLOSE          YES,NO,REXX exit (indexes only)
==MSG>    TRACKMOD           YES,NO,REXX exit (table spaces only)
==MSG>    DCAPTURE           NONE,CHANGES,REXX exit (tables only)
==MSG>    FREEPG             n,REXX exit (table spaces and indexes)
==MSG>      TSFREEPG         n,REXX exit (table spaces only)
==MSG>      IXFREEPG         n,REXX exit (indexes only)
==MSG>    PCTFREE            n,REXX exit (table spaces and indexes)
==MSG>      TSPCTFREE        n,REXX exit (table spaces only)
==MSG>      IXPCTFREE        n,REXX exit (indexes only)
==MSG>    TSPCTFUPD          n,REXX exit (table spaces only)
==MSG>    LOCKMAX            n,SYSTEM,REXX exit (table spaces only)
==MSG>    ERASE              YES,NO,REXX exit (table spaces and indexes)
==MSG>      TSERASE          YES,NO,REXX exit (table spaces only)
==MSG>      IXERASE          YES,NO,REXX exit (indexes only)
==MSG>    RESONDROP          YES,NO,REXX exit (tables only)
==MSG>    EDITPROC           string,REXX exit (tables only)
==MSG>    VALIDPROC          string,REXX exit (tables only)
==MSG>    TSPARTS            n,REXX exit (table spaces)
==MSG>    LOGGED             YES,NO,REXX exit (table spaces only)
==MSG>    LOCKSIZE           TABLE,TABLESPACE,PAGE,ROW,LOB,ANY,REXX exit
==MSG>                                        (table space only)
==MSG>    MAXROWS            n,REXX exit (tables only)
==MSG>    GBPCACH            SYSTEM,CHANGED,ALL,NONE,REXX exit
==MSG>                                        (table spaces and indexes)
==MSG>    TSGBPCACH          SYSTEM,CHANGED,ALL,NONE,REXX exit
==MSG>                                        (table spaces only)
==MSG>    IXGBPCACH          SYSTEM,CHANGED,ALL,NONE,REXX exit
==MSG>                                        (indexes only)
==MSG>    VOLATILE           YES,NO,REXX exit (tables only)
==MSG>    APPEND             YES,NO,REXX exit (tables only)
==MSG>    PADDED             YES,NO,REXX exit (indexes only)
==MSG>    COPY               YES,NO,REXX exit (indexes only)
==MSG>    MEMCLUS            YES,NO,REXX exit (table spaces only)
==MSG>    FIELDPROC          string,REXX exit (tables only)
==MSG>    INSALGO            n (0-2),REXX exit (table spaces only)
==MSG>    SGKEYLABL          string,NO,NOKEYLABEL,REXX exit (stogroup only)
==MSG>    TBKEYLABL          string,NO,NOKEYLABEL,REXX exit (tables only)
==MSG>
```

*Figure 23. **Edit Compare Masks (GOCEDIT)** panel, part 2*

```
==MSG>    Verification mask Syntax:
==MSG>       VER,Field:operand,value(,values),RC=x
==MSG>           or
==MSG>       VER,rexxField:REXX(exitproc,parm1,parm2,...parmn)
==MSG>      where:
==MSG>     Field:         Same fields used by overwrites
==MSG>     RexxField      Can be one of three options:
==MSG>                    1. same fields used by overwrites
==MSG>                    2. special REXX only field, OBJNAME or TSPARTS
==MSG>                    3. two char object type code listed below:
==MSG>                         code   Object type       Catalog record
==MSG>                          SG    Storage group      SYSSTOGROUP
==MSG>                          DB    Database           SYSDATABASE
==MSG>                          TS    Table space        SYSTABLESPACE
==MSG>                          TB    Table              SYSTABLES
==MSG>                          IX    Index              SYSINDEXES
==MSG>                          TG    Trigger            SYSTRIGGERS
==MSG>                          FK    Foreign Key        SYSRELS
==MSG>                          PK    Primary key        SYSTABCONST
==MSG>                                                   TYPE = P
==MSG>                          CK    Check Constraint   SYSCHECKS
==MSG>                          UQ    Unique COnstraint  SYSTABCONST
==MSG>                                                   TYPE = U
==MSG>                          DT    Data type          SYSDATATYPES
==MSG>                          FU    Function           SYSROUTINES
==MSG>                          SP    Procedure          SYSROUTINES
==MSG>                          SQ    Sequence           SYSSEQUENCES
==MSG>                          SY    Synonyms           SYSSYNONYMS
==MSG>                          AL    Alias              SYSTABLES
==MSG>                          VW    View               SYSVIEWS
==MSG>                          GV    Global variable    SYSVARIABLES
==MSG>                          TP    Table part         SYSTABLEPART
==MSG>                          IP    Index part         SYSINDEXPART
==MSG>                          FL    Fields             SYSFIELDS
==MSG>     Operand:        EQ     - Equal
==MSG>                     NE     - Not equal
==MSG>                     GT     - Greater than
==MSG>                     LT     - Less than
==MSG>                     LIST   - list of values
==MSG>                     RANGE  - range of values from two input values
==MSG>     value:          same values as overwrite values
==MSG>     RC=:            return code if expression is not met
==MSG>     x:              return code value - 0,4,8,12
==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 DBNAME) or
==MSG>       a variable with numeric/string value (such as BPOOL= 'BP1').
==MSG>       + in col 72 indicates continuation of Rexx exit on next line
==MSG>     - To support/migrate DB2V8 masking input,OWNER,TBOWNER and
==MSG>       IXOWNER will mask both owner and schema fields.SCHEMA,
==MSG>       TBSCHEMA and IXSCHEMA will be applied to schema fields only.
==MSG>     - For DB2 synonyms, apply DB2 APAR PM42910 in DB2 V9 NFM and
==MSG>       above and then use schema as the qualifier. SYNOWNER is
==MSG>       migrated into SYNSCHEMA. Use SYNSCHEMA instead of SYNOWNER.
==MSG>     - SINGLECH format is SINGLECH:<character>Ý,<escape character>¨
==MSG>       where the single character in mask specification represents
==MSG>       any character at that position. If the specified escape
==MSG>       character precedes the specified single character, then the
==MSG>       single character is treated as literal.
==MSG>     - The view, alias and synonym masks (both name and
==MSG>       schema/owner) will only apply to the CREATE statement for
==MSG>       these objects (e.g. VWNAME only valid for CREATE VIEW).
==MSG>       All other usages of these names and schemas are vague and
==MSG>       can refer also to table names and schemas.  These other
==MSG>       usages can only be masked by TBNAME for name and TBSCHEMA
==MSG>       for schema; therefore, it is recommended to use both VWNAME
==MSG>       and TBNAME if view names are being changed for both CREATE
==MSG>       VIEW statement and SQL that uses this view.
==MSG>     - Use caution when specifying mask field SEGSIZE. This mask
==MSG>       field might cause changes to the table space type. For
==MSG>       example, specifying the SEGSIZE mask might convert a
==MSG>       partitioned table space to a range-partitioned universal
==MSG>       table space (UTS). If a table in a UTS has a partitioned
==MSG>       index and the partitioned index needs to be recreated, DB2
==MSG>       might generate SQLCODE=-662 during execution.
```

*Figure 24. **Edit Compare Masks (GOCEDIT)** panel, part 3*

```
==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>    - Verification mask checks attributes using expression given
==MSG>      and if the expression is false, return code of value given
==MSG>      will be issued.  If return code is greater than 4,
==MSG>      processing will fail after all objects are processed and
==MSG>      error messages will be in VALOUT file.
==MSG>    - OBJNAME is a special verification mask type that only is
==MSG>      allowed with REXX exec syntax.  OBJNAME will provide three
==MSG>      arguments to REXX exec, object type, object name and object
==MSG>      schema.
==MSG>    - CK mask will not be triggered for DB2 generated checks like
==MSG>      DB2_GENERATED_CHECK_CONSTRAINT_FOR_SYSTEM_TIME and
==MSG>      DB2_GENERATED_CHECK_CONSTRAINT_FOR_BUSINESS_TIME
==MSG>    - Masks should start in column 1.
==MSG>  Mask examples:
==MSG> OWNER:ABC*,DEF*
==MSG> NAME:PRE*,NPRE*
==MSG> XMLSCHID:PO1,PO2
==MSG> WLMENV:WLM33,WLM44
==MSG> LOCATION:LOC3*,LOCT*
==MSG> SETPATHSC:SYSIBM,SYSFUN
==MSG> SINGLECH:_
==MSG> SINGLECH:_,+
==MSG>
==MSG>  Object-specific mask examples:
==MSG> TBSCHEMA:CREATOR1.TB2:CREATOR1,NEW_CRE1
==MSG> IXNAME:IXOWN*.IX3*:IX3*,IX4*
==MSG> IXBPNAME:IXOWN1.INDX2:BP1,BP3
==MSG>
==MSG>  Overwrite examples:
==MSG> COMPRESS:MYDB*.MYTS*,YES
==MSG> SEGSIZE:MYDB*.MYTS*,8
==MSG> DSSIZE:MYDB*.MYTS*,4G
==MSG> PRIQTY:*.*,REXX(MYPRIQTY,DBNAME='MYDBTEST')
==MSG> TSPRIQTY:MYDB*.MYTS*,30
==MSG> IXPRIQTY:MYCR*.MYIX*,25%
==MSG> IXSECQTY:MYCR*.MYIX*,REXX(MYSECQTY,IXNAME,IXCREATOR,PCT=20%)
==MSG> DEFER:USER001.*IXNAME,NO
==MSG> DEFINE:DBNAME*.*TSPC,REXX(MYDEFINE,DEFINE='YES')
==MSG> HASHSPC:TBCREATOR.MYTBNAME,100M
==MSG> TBINLOBL:TBCREATOR.MYTBNAME.COLNAME,16000
==MSG> DTINLOBL:DTCRE*.DTNAME*,16000
==MSG> IXCLOSE:MYCR*.MYIX*,NO
==MSG> AUDIT:MYDB*.MYTB*,CHANGES
==MSG> TRACKMOD:MYDB*.MYTS*,NO
==MSG> DCAPTURE:TBCRE*.MYTB*,NONE
==MSG> FREEPG:ABC*.DEF*,6
==MSG> IXPCTFREE:IXSCH1.IXNAME1,9
==MSG> LOCKMAX:DBTEST2.TSTEST2,SYSTEM
==MSG> TSERASE:DBTEST1.TSTEST1,NO
==MSG> RESONDROP:TBCRE*.MYTB*,NO
==MSG> TSPCTFUPD:DB1.TS1,25
==MSG> INSALGO:DB1.TS1,2
==MSG> SGKEYLABL:SG1,DB2SYS_KEY01
==MSG> SGKEYLABL:SG1,NOKEYLABEL
==MSG> SGKEYLABL:SG1,NO
==MSG> TBKEYLABL:TBCRE*.MYTB*,DB2SYS_KEY02
==MSG> TBKEYLABL:TBCRE.MYTB,NOKEYLABEL**
==MSG> TBKEYLABL:TBCRE.MYTB,NO**
==MSG>
==MSG>  Verification mask examples:
==MSG> VER,COMPRESS:EQ,YES,RC=4
==MSG> VER,EDITPROC:NE,PROC1,RC=8
==MSG> VER,TSPARTS:LT,65,RC=8
==MSG> VER,PCTFREE:GT,20,RC=8
==MSG> VER,SEGSIZE:LIST,4,8,12,RC=8
==MSG> VER,PCTFREE:RANGE,0,5,RC=4
==MSG> VER,OBJNAME:REXX(OBJTST)
==MSG> VER,SEGSIZE:REXX(SEGTST,MYSEGSZ)
==MSG> VER,MEMCLUS:EQ,NO,RC=8
==MSG> VER,FIELDPROC:NE,FLDPROC1,RC=8
==MSG> VER,IXCOMPRES:EQ,YES,RC=4
==MSG> VER,TSCOMPRES:REXX(COMPTST,DBNAME,NAME,COMPRESS)
==MSG> VER,TSPCTFUPD:RANGE,10,20,RC=8
==MSG> VER,INSALGO:RANGE,0,2,RC=8
==MSG> VER,TBKEYLABL:NE,DB2SYS.KEY02,RC=8
```

*Figure 25. **Edit Compare Masks (GOCEDIT)** panel, part 4*

```
==MSG>
==MSG>  Verification object type mask examples:
==MSG> VER,IX:REXX(VERIX,TBCREATOR,TBNAME,NAME)
==MSG> VER,DB:REXX(VERDB,NAME,CREATOR,BPOOL)
==MSG>
==MSG>  Syntax for info about renamed objects/columns:
==MSG>    renameobj:old-name,new-name
==MSG>    RENAMECOL:table-name.old-colname,new-colname
==MSG>      ( + in col 72 indicates continuation on next line col 1)
==MSG>  renameobj:
==MSG>    RENAMEDB, RENAMETS, RENAMETB, RENAMEIX,
==MSG>    RENAMEGV
==MSG>  Examples:
==MSG>    RENAMETB:OLDOWNER.OLDNAME,NEWOWNER.NEWNAME
==MSG>    RENAMECOL:OWNERA.MYTB.OLDCOLNAME,NEWCOLNAME
==MSG>
****** ***************************** Bottom of Data ****************************
```

*Figure 26. **Edit Compare Masks (GOCEDIT)** panel, part 5*

4. Specifying ignores

When you compare objects, you can optionally specify ignores and ignore changes.

An *ignore* is one or more fields that Object Comparison Tool is to ignore when comparing Db2 catalog records. Specify ignores when the source and target objects have differences, but you do not want these differences to result in changes to the target.

An *ignore change* is a specified change to an object that was previously reported in a compare result and you want ignored. Ignore changes are reported, but no SQL statements are generated for the changes.

Before you begin

If you want to specify an ignore change, you must have saved compare results. Optionally, you can define an ignore change specification in advance; see “Creating ignore changes specifications” on page 122.

About this task

When specifying ignores in Db2 Object Comparison Tool, you can define the ignores either in the Change Management (CM) database or in a data set. For more information about ignores, see Ignore fields (IBM Db2 Administration Tool for z/OS 13.1.0).

Procedure

To specify ignores to be used during the compare process:

1. On the **DB2 Object Comparison Tool Menu (GOCMENU)** panel, specify option 4, and press Enter.
   The **Specify Compare Ignores (GOC4)** panel 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 . . . .                              > (? 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 27. **Specify Compare Ignores (GOC4)** panel*

Note: If Change Management is not enabled, only the **Data Set Name** and **Edit Ignore Fields Specification** fields are displayed.

2. Complete one or both of the following tasks, depending on the ignores that you want to use:
   - "Specifying ignore fields" on page 74
   - "Specifying ignore changes" on page 77

Related information

Ignore fields (IBM Db2 Administration Tool for z/OS 13.1.0)

Specifying ignore fields

Procedure

1. On the **Specify Compare Ignores (GOC4)** panel, specify the following information in the **Ignore Fields Specification** section:

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

*Figure 28. Ignore Fields Specification section on the **Specify Compare Ignores (GOC4)** panel*

- Specify one of the following sources that contains (or will contain) the ignore fields specifications:

  CM table
  Specify **Owner** and **Name** values for the ignore. This name identifies a row in the Change Management repository table . That row contains (or will contain) the ignore fields that you want to use for the comparison operation. You can specify either an existing name to identify an existing row in the table or a specify a new name to create a new row in the table.

  Data set
  Specify a value for **Data set name**. The data set 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.

If CM is not enabled, you must specify a data set. If both a CM table and a data set are specified, the information in the data set field is used.

- In the **Edit Ignore Fields Specification** field, specify whether you want to edit the ignore fields.

2. Press Enter.

   One of the following panels is displayed:

   - If you specified that you do not want to edit the ignore fields, the **DB2 Object Comparison Tool Menu (GOCMENU)** panel is displayed. You have completed specifying ignore fields. You can skip the rest of this procedure.
   - If you specified that you wanted to edit the ignore fields and your source is a data set or an existing row in the CM repository table , the **Specify Ignore Fields : Objects (GOCCI)** panel is displayed. Skip to step “4” on page 75.
   - If you specified that you wanted to edit the ignore fields and your source is a new row in the CM repository table, the **Insert Ignore (ADB2C22)** panel is displayed.

3. On the **Insert Ignore (ADB2C22)** panel, create a new ignore specification by completing the following steps:

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


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

*Figure 29. **Insert Ignore (ADB2C22)** panel*

   a) Optional: In the **Comment** field, specify a description of the ignore.

   b) Press Enter to insert a row into the Change Management repository table for this new set of ignore fields.

   c) Exit (PF3) to display the **Specify Ignore Fields : Objects (GOCCI)** panel.

4. On the **Specify Ignore Fields : Objects (GOCCI)** panel, edit the ignore fields as needed:

   This panel shows the objects and the fields within the objects, if any, that are to be ignored. In the following example, several objects have ignore fields defined. For example, for SYSDATABASE, the fields CREATOR, STGROUP, BPOOL, and INDEXBP are to be ignored.

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

 Valid line commands are:
  U - Update Ignore Fields

 Select Object          Ignore Fields                                          Qualifier Name
        *               *                                                      *         *
 ------ --------------- ------------------------------------------------------ ---------
---------------
        GENERIC         None
        SYSCHECKS       None
        SYSCOLUMNS      COLTYPE,LENGTH,SCALE,DEFAULT,DEFAULTVALUE
        SYSCONTROLS     None
        SYSDATABASE     CREATOR,STGROUP,BPOOL,INDEXBP
        SYSDATATYPES    None
        SYSENVIRONMENT  APPLCOMPAT
        SYSFIELDS       None
        SYSINDEXES      None
        SYSINDEXPART    PQTY,SQTY,SECQTYI
        SYSKEYS         None
        SYSPACKAGE      None
        SYSPARMS        None
        SYSRELS         None
        SYSROUTINES     None
        SYSSEQUENCES    None
        SYSSTOGROUP     VCATNAME
        SYSTABLEPART    PQTY,SQTY,SECQTYI
        SYSTABLES       STATUS,LABEL                                           SYSADM   LRC*87
        SYSTABLES       LABEL,CHECKRID                                         SYSADM   LRC*88
        SYSTABLES       AUDITING
        SYSTABLESPACE   None
        SYSTRIGGERS     None
        SYSVIEWS        None
        SYSVOLUMES      VOLID
        XMLMODIFIER     None
```

*Figure 30. **Specify Ignore Fields : Objects (GOCCI)** panel*

This panel contains the following fields:

Object
The name of the Db2 catalog table that describes the object or part of the object.

Ignore Fields
The field or fields to ignore for this catalog table.

Qualifier
The qualifier of the specific object or objects to be ignored.

Name
The name of the specific object or objects to be ignored.

Tip: Use caution when specifying ignores. If possible, use the generic ignore field specifications. These specifications provide for some common sets of fields that are often intentionally different on source and target systems. For more information about generic ignores, see Generic ignores (IBM Db2 Administration Tool for z/OS 13.1.0).

a) Issue the U (update) line command for the appropriate object (catalog table or the generic object), and press Enter.

The **Select Ignore Fields for object (GOCCIF)** panel is displayed for the selected object:

```
GOCCIF --------- Select Ignore Fields for SYSINDEXPART ----- Row 1 to 18 of 22

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

Select Fields                Action   Qualifier Name
       *                     *        *         *
------ --------------------- -------- --------> ----------------------------->
       AVGKEYLEN
       CREATEDTS
       DSNUM
       EXTENTS
       FREEPAGE
       GBPCACHE
       INDEXTYPE
       LEAFFAR
       LEAFNEAR
       LIMITKEY
       OLDEST_VERSION
       PARTITION
       PCTFREE
       PQTY
       PSEUDO_DEL_ENTRIES
       RBA_FORMAT
       SECQTYI
       SPACEF
```

*Figure 31. **Select Ignore Fields for object (GOCCIF)** panel*

b) Select the fields that are to be ignored by using the S line command, and press Enter.

You can further limit an ignore field to certain objects by specifying values in the **Qualifier** and **Name** columns with wildcard characters in any field. You can modify the qualifier and name of a field only when that row is selected. To apply multiple *qualifier.name* combinations to a field, issue the R line command to create additional rows for hat field.

c) Exit (PF3) back to the **Specify Ignore Fields : Objects (GOCCI)** panel.

d) Exit (PF3) back to the **DB2 Object Comparison Tool Menu (GOCMENU)** panel.

Related tasks

Specifying ignore changes

Specifying ignore changes

Procedure

1. On the **Specify Compare Ignores (GOC4)** panel, specify one of the following sets of information in the **Ignore Changes Specification** section:

   a. If you want to use a defined ignore changes specification, specify the owner and name of the specification and whether you want to edit the specification in the following fields:

```
Ignore Changes Specification:
  Owner . . . .                              > (? to look up)
  Name  . . . .                              > (? to look up)
  Edit Ignore Changes Specification  . . . NO  (Yes/No)
```

*Figure 32. Ignore Changes Specification section on the **Specify Compare Ignores (GOC4)** panel*

   b. If you want to use a saved compare result, specify that you want to display the saved compare result and an owner and name for the result in the following fields:

```
Display using a saved compare result . . YES (Yes/No)
  Saved Compare Results:
    Owner . . . .                          > (? to look up)
    Name  . . . .                          > (? to look up)
```

*Figure 33. Ignore Changes Specification section on the **Specify Compare Ignores (GOC4)** panel*

Restriction: You cannot specify both a and b; you can specify either an ignore changes specification or a saved compare result.

2. Press Enter.
3. Take one of the following actions, depending on which panel is displayed:
   - If you specified that you do not want to edit or display the ignore changes, the **DB2 Object Comparison Tool Menu (GOCMENU)** panel is displayed. You have completed specifying ignore changes. Skip the rest of this procedure.
   - If you specified that you want to edit the ignore changes specification, the **Ignored Changes List (ADBPCICL)** panel is displayed. Modify the specification as needed. See step "3" on page 124 in "Modifying ignore changes specifications" on page 123.
   - If you specified that you want to display the saved compare results, the **Compare Report (ADBPCRR)** panel is displayed. Select the changes that you want to ignore. See step "4" on page 122 in "Creating ignore changes specifications" on page 122.

What to do next

Generate a compare batch job.

Related tasks

Specifying ignore fields

5. Generating a compare batch job

A *compare batch job* is a JCL job that performs the requested comparison.

Before you begin

Before you can generate a compare job, you must complete the following tasks:

- "1. Specifying source objects" on page 48
- "2. Specifying target objects" on page 60

You can tell whether the source and target are specified by looking at the **Specification Status** column on the **DB2 Object Comparison Tool Menu (GOCMENU)** panel.

You can optionally complete the following additional tasks before generating a compare batch job:

- "3. Specifying compare masks" on page 63
- "4. Specifying ignores" on page 73

If you want to use previous compare batch job selections that were saved as a dialog, restore that dialog before you begin. See "Managing and restoring dialogs" on page 111.

Procedure

To generate a compare batch job:

1. On the **DB2 Object Comparison Tool Menu (GOCMENU)** panel, specify option 5, and press Enter.

   The **Generate Compare Jobs (GOC5)** panel is displayed.
2. If want to use Change Management (CM) or do a multi-target import, skip the remaining steps and complete one of the following procedures instead:

- "Generating a compare batch job to make changes through Change Management" on page 100
- "Generating a compare batch job for a multi-target import" on page 102

3. On the **Generate Compare Jobs (GOC5)** panel, specify a value for each option, and press Enter.

   For information about each option on this panel, see "Compare job options" on page 80.

   Tip: Set **Save compare results** to YES if you want to analyze data about the comparison, ignore changes, or increase the efficiency of subsequent comparisons. The saved compare results contain information about objects that were part of the comparison, including detected differences, changes to make, and how those changes are to be implemented. You can save the compare results only for tables, indexes, global variables, and distinct data types.
4. If the **Change Management Prompt (ADB2CMRO)** panel is displayed (because CM is enabled and optional for your ID), specify whether to use CM, and press Enter.

   If you use CM, complete the steps in "Generating a compare batch job to make changes through Change Management" on page 100 instead.
5. Complete the requested input on any subsequent panels that are displayed.

   One or more of the following panels might be displayed depending on the compare job options that you selected. For more information about one of these panels, see the related job option description.

Panel	Option on Generate Compare Jobs (GOC5) panel
Save Compare Results (ADB2C22) panel	"Save compare results" on page 84
Specify Compare Reporting Options (GOC5RO) panel	"Change reporting options" on page 83
Specify Data Set Name for Apply Jobs (GOC5AJ) panel	"Generate apply jobs" on page 86
Specify Work Statement List Data Set (ADB2WLDA) panel	"As work statement list" on page 89
Specify Work Statement List (ADB27WLD) panel	"As work statement list" on page 89
Specify Work Statement List Data Set (GOC5WL) panel	"As work statement list" on page 89
Specify Job Parameters (ADB2W1R) panel	"As work statement list" on page 89
REBIND options (ADBPREBO) panel	"REBIND options" on page 95
DB2 Object Compare Warning (GOCGCMPW) panel	"Generate apply jobs" on page 86

6. Edit the generated JCL job as needed and submit it to run the comparison.

   For information about the Object Comparison Tool parameters in the generated JCL job, see "Parameters in the generated compare batch job" on page 106.

What to do next

Tip: Consider saving your current compare batch job selections for later use. See "Saving dialogs" on page 110.

Compare job options

When you generate a compare batch job, you can specify a number of options to control the behavior of the comparison operation and job. These options are listed on the **Generate Compare Jobs (GOC5)** 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)
   Drop FKs not in source  . . NO        (Yes/No)
   Suppress DROP of columns  . NO        (Yes/No)
   Suppress adding columns . . NO        (Yes/No)
   Run SQLID . . . . . . . . .           (Blank, an SQLID, or <NONE>)
   Object Grantor  . . . . . .           (Blank or an SQLID)
   Run Validate. . . . . . . . V         (Validate, None)
   Allow implicit drop of
      excluded objects . . . . NO        (Yes/No)
   Enable auth-switching . . . YES       (Yes/No)
   Disable REORG optimization  YES       (Yes/No)
   Scope Warning Messages  . . 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)
   Allow deferred restart  . . NO        (Yes/No)
   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)
        Embed IFF into WSL  . . NO       (Yes/No)
     Use customized util opts. YES       (Yes/No)
     Content of apply job(s) . ALL       (All, DDL)
     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 START and RESTART values:
     For sequence object . . .          (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/Min/None)
       Run REBIND  . . . . . . M         (Mandatory, All relevant, None)
         REBIND options  . . . YES       (Yes/No)

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

*Figure 34. **Generate Compare Jobs (GOC5)** panel*

The options on this panel are described in the following sections:

Worklist information:

Worklist name
Specify the name of the work statement list to use.

This name is also used as the middle qualifier in the names of the work data sets that are created for the job. The prefix for these 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 for data sets** value, the **Worklist name** value, 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** value is also used as a name for the Db2 Administration Tool SQL or DDL executor, which has a checkpoint facility. The **Worklist name** value is used as a key to the checkpoint table. Use a unique name for each comparison that you run.

Related information:

Work statement lists (IBM Db2 Administration Tool for z/OS 13.1.0)

Compare options:

Suppress DROP of objects
Specify whether the compare process is to drop objects that are in the target, but not in the source. Specify Yes to prevent the compare process from dropping any target objects.

Regardless of the value that you set for this option, 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).

Also, if DROP statements are part of the source DDL, objects are dropped regardless of the value specified for this option.

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, all of the LOB objects are kept.

Drop FKs not in source
Specify whether the compare process is to drop from the target table any foreign keys that are not specified in the corresponding source table. If you specify No, the drop behavior is determined by **Suppress DROP of objects** field. The default is No.

Suppress DROP of columns
Specify whether the compare process is to drop columns that are in the target tables but not in the source table. Specify Yes to prevent the compare process from dropping any columns.

Suppress adding columns
Specify whether the compare process is to add columns to the target. Specify Yes to prevent columns in the source from being added to the target. This option is useful if you have extra columns on your source that you do not want added to your target.

Run SQLID
Specify a valid SQL ID to use when creating, dropping, or altering objects. This ID is typically an administrative SQL ID whose only privileges are to create objects.

If a value is specified, this SQL ID becomes the owner of the databases and table spaces. If the specified SQL ID is different from the current owner, the databases and table spaces (and all dependent objects) are dropped and recreated to change the owner.

If you enter <NONE>, a SET CURRENT SQLID statement is not generated in the DDL.

If you leave this field blank, a SET CURRENT SQLID statement is generated in the DDL before each object that is created. Where possible, the SQL ID that was originally used to create the object is used in the SET statement.

Object Grantor
Specify an SQL ID to use in SET CURRENT SQLID statements that precede GRANT statements.

If an SQL ID is specified for this option, but the **Run SQLID** option is set to <NONE>, no SET CURRENT SQLID statements are generated to precede GRANT statements.

Run Validate
Specify whether to perform consistency checking. Consistency checking verifies that for all of the primary objects in the source DDL, any dependent objects exist. These dependent objects must exist in the source DDL or the target catalog.

When you specify V or `Validate`, 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.
- If the index in a CREATE INDEX statement is a clustering index, a clustering index does not already exist.
- 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.
- The number of index partitions matches the number of table space partitions.

Object Comparison Tool also checks that the dependent objects exist if the following statements are generated:

- CREATE TRIGGER
- CREATE VIEW
- CREATE MQT
- CREATE INDEX
- ADD FOREIGN KEY

When you request consistency checking, a consistency checks report (ADB2WVL) is generated. If a check fails, a message is written to the report with a return code of 8.

Note: For native stored procedures, even if validation is successful, the existence of the object in the native stored procedure body cannot be known at procedure run time (or during procedure call).

Allow implicit drop of excluded objects
Specify whether excluded objects can be dropped implicitly.

If you specify `Yes`, excluded objects can be dropped if needed and are then recreated according to the target definition. If you specify No and an excluded object needs to be dropped, an error message is displayed and the compare fails. No is the default.

Enable auth-switching
Specify whether to 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.

This field is visible only if the authorization switching facility is enabled for the subsystem during the customization process.

Disable REORG optimization
Specify whether you want to disable REORG optimization. REORG optimization reduces the number of REORG utility statements that are issued, and thus the number of times that your system halts. The default value is No.

Scope Warning Messages
Specify whether to issue a warning message in the case where the target of an object comparison operation is automatically selected, and the source is not a table space. This message warns that objects that exist only in the target might be dropped.

If you specify Yes, message ADB7353 is issued for this situation. This message is issued regardless of the value of the **Suppress DROP of objects** option.

Reporting options:

Change reporting options
Specify whether you want to change the options for reports.

If you specify Yes, the following panel is displayed after you press Enter. You can change the reporting options on this panel:

```
GOC5RO  --------- Specify Compare Reporting Options    ---------- 12:20

 Report options for Compare:
   Only changed objects  . . . YES      (Yes/No)
   Ignore fields:
     User specified  . . . . . YES      (Yes/No)
     System generated  . . . . YES      (Yes/No)
     Object specific . . . . . YES      (Yes/No)
   Translation masks   . . . . YES      (Yes/No)
   Summary report  . . . . . . YES      (Yes/No)
   Object count report . . . . YES      (Yes/No)
   Conversion report . . . . . YES      (Yes/No)
```

*Figure 35. **Specify Compare Reporting Options (GOC5RO)** panel*

Only changed objects
Specify whether the detailed report is to include only those objects that have changed.

Ignore fields:

User specified
Specify whether the report is to include the names of user-specified ignore fields.

System generated
Specify whether the report is to include the names of system ignore fields.

Object Specific
Specify whether the report is to include the names of fields that are ignored for specific objects.

Related information:

“4. Specifying ignores” on page 73

Translation masks
Specify whether the report is to include the translation masks that are used by the compare job.

Related information:

“Translation masks” on page 65

Summary report
Specify whether the report is to include a summary, which consists of one line per object.

Object count report
Specify whether the report is to include statistics of compared and changed objects.

Conversion report
Specify whether to report expected conversion problems for tables when a change is run.

The following example shows the corresponding strings for the parameters that are passed to step T03COMP PGM=GOC2CMP if you specify YES for the fields on panel GOC5RO. Specifying Yes for both **User specified** and **System generated** results in REPIGALL being used as the parameter.

```
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  . . : REPCHG                                   |
  Ch|    Ignore fields:                                                       |
    |      User specified  . . . . : REPIGUSR                                 |
    |      Object Specific . . . . : Yes        (Yes/No)                      |
  Da|      System generated  . . . : REPIGSYS                                 |
    |    Translation masks   . . . : REPMASK                                  |
    |    Summary report  . . . . . : REPSUM                                   |
  Op|    Object count report . . . : REPCOUNT                                 |
    |    Conversion report . . . . : REPCONV                                  |
    |                                                                         |
    |                                                                         |
    |                                                                         |
    '-------------------------------------------------------------------------'
```

*Figure 36. Example of the **Specify Compare Reporting Options (GOC5RO)** panel with REPIGALL used as the PARM options.*

Save compare results
Specify whether compare results are to be saved.

Restriction: Compare results are saved for only the following objects:

- tables
- indexes
- global variables
- distinct data types

If you specify YES, the following panel prompts you to specify a name for the saved result:

```
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 37. **Save Compare Results (ADB2C22)** panel*

Data set information:

PDS for jobs
Specify the name of the partitioned data set (PDS) where the compare jobs are to be generated.

Prefix for data sets
Specify the prefix to be used for data sets that are allocated by the batch job. For example, UNLOAD, DDL, and LOAD data sets can be allocated.

CHANGES file data set name
Specify the name of the data set to store the changes that are output by the compare job. This data set is used primarily for changes that are to be imported to Change Management. To generate these changes for Change Management, set **Generate apply jobs** to Change.

If this data set is partitioned, it must be preallocated, and you must specify a member name in the **Member name** field.

If you preallocate this data set, the data set must meet the following requirements:

- For delta changes, this data set must be either fixed length with an LRECL of 80 or variable length with an LRECL of 16384.
- For changes, this data set must be variable length with an LRECL of 16384.

Member name
If the CHANGES data set is partitioned, specify a member name.

Options:

Generate online
Specify whether the compare process is to be run online.

If you specify Yes, the compare process runs immediately when you press the Enter key after specifying the compare job options. If **Generate apply jobs** and **As work statement list** are also set to Yes, the work statement list is created online but is not run. You can run the work statement list later.

If you specify No, a batch job is generated. You can submit this batch job later to perform the compare process in the background. The default value is No.

When the compare process is run in batch, messages are placed in the SYSPRINT data set. You can override this output data set by using the TU command to define the ADBWORK template. If you do not specify Yes for the **Generate templates** option and define ADBWORK, the default data set name, *prefix*.*wsl*.SYSPRINT, is used.

Restriction: This **Generate online** function is not available when comparing multiple sources and targets.

Related information:

"Running a work statement list to apply changes" on page 127
"TU - Specify TEMPLATE usage" on page 97

Single compare job
Specify whether all job steps are to be executed in one job.

Use separate jobs to run the source extraction on a system other than the target system.

If you specify No, up to four jobs are generated for the following actions:

- Extract the source if the source is DDL or the Db2 catalog
- Extract the target if the target is DDL or the Db2 catalog
- Compare the source and target
- If **Generate apply jobs** is set to Yes, generate apply jobs or register job when Change Management is enabled

If you specify Yes, you must specify a member name in the **Member name** field.

Member name
If you requested a single job, specify the name of the member where the compare job is to be generated. The default is COMPARE.

Allow deferred restart

Specify whether the generated compare job is to support deferred step restart. You can specify one of the following values:

YES

Generated compare JCL will not include backward references to previous steps so that deferred step restart is possible. This option is ignored if Generate online is YES or Single compare job is NO.

NO

Generated compare JCL will include backward references when allocating temporary data sets. Therefore, deferred restart is not possible without changing these backward references. NO is the default.

Related information:

Backward references (z/OS 2.5.0)

Generate apply jobs

Specify whether to generate jobs to apply the changes that were found during the comparison to the target objects.

Instead of generating apply jobs, Object Comparison Tool can also generate a work statement list or register a change in the Change Management database on the target subsystem. You can then use the apply jobs, work statement list, or change to apply changes to the target object. This process is shown in Figure 1 on page 11.

The **Generate apply jobs** function uses the following input:

- The changes file from the compare operation, which contains the following information:
  - The DROP, CREATE, and ALTER statements
  - The UNLOAD requests
  - Table space information records, which allow Object Comparison Tool to determine the size of the UNLOAD requests

  Restriction: Do not attempt to import a CHANGES file that is 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, which contains the variables that were specified in the panels.

You can specify one of the following values:

Yes

Generate apply jobs.

If you request the generation of apply jobs, run the Object Comparison Tool dialog connected to the target Db2 subsystem to pick up the correct libraries for use in the apply jobs. Alternatively, if the Object Comparison Tool 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, you also have the option to create a work statement list, use the utility options, select the unload method, and generate templates.

If you specify Yes and **As work statement list** = NO, the following panel prompts you for a data set name for the apply jobs:

```
GOC5AJ  --------- Specify Data Set Name for Apply Jobs ---------- 12:22

Enter/verify the following:
 Data Set Name  ===> ADM001.COMPARE.D97220
```

*Figure 38. **Specify Data Set Name for Apply Jobs (GOC5AJ)** panel*

No

Do not generate jobs to apply the changes.

Change
Generate a delta change in the CHANGES file that can later be imported to Change Management. (The CMDELTA parameter for GOC2CMP). No apply jobs or work statement list is generated.

If you specify Change, the compare process is said to run in *CMDELTA mode*.

The **Generate apply jobs** function runs as an EXEC (ADBGAJOB) in a TSO/ISPF batch job and uses ISPF skeletons to generate the apply jobs or work statement list.

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 can result. In this case, Object Comparison Tool issues the following warning message, which gives you 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 39. **DB2 Object Compare Warning (GOCGCMPW)** panel*

Generate one job
Specify whether to generate a single apply job.

Restriction: The **Generate one job** option does not apply when **As work statement list** is set to Yes.

You can specify one of the following values:

Yes
A single apply job is generated. If the number of steps exceeds a maximum limit of 255, more than one job is generated.

If you specify Yes, specify a value in the **Member prefix** field.

No
Multiple jobs are generated. Specifically, Object Comparison Tool produces the following apply jobs:

T10U*nnnn*
UNLOAD jobs, which contain the following steps:

- Step 1 issues a Db2 command to place the table space in read-only status.
- Step 2 deletes the SYSREC and SYSPUNCH data sets, if they already exist.
- Step 3 unloads the given table.
- Additional 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, evaluate 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, such as those created from DDL, do not represent the same information that is found in the Db2 catalog.

T20DROP
DROP job. This job includes statements that need to be executed before objects are created. It can include DROP, RENAME, and ALTER statements. This job contains DD statements that reference 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. It can also contain ALTER SQL statements.

If an inline utility needs to be run between DDL statements, the T30CREAT job is split into multiple jobs, named T30C*nnnn*.

T40STOP
STOP job. This job stops page sets. It can contain other SQL and utility statements.

T50ALTER
ALTER job. This job can contain CREATE and ALTER SQL statements.

If an inline utility needs to be run between DDL statements, the T50ALTER job is split into multiple jobs, named T50A*nnnn*.

T60START/T61START
START jobs. These jobs start page sets. The T61START job is for clone objects.

T70R*nnnn*
RELOAD jobs.

T71R*nnnn*
REORG jobs. These REORG jobs remove REORG-pending conditions. All REORG statements are combined into a single job if SHRLEVEL CHANGE is specified and the mapping table name is provided.

T72REBLD
REBUILD jobs.

T8RO*nnnn*
REORG jobs. These REORG jobs fully implement the effects of the changes (for example, space parameter changes). All REORG statements are combined into a single job if SHRLEVEL CHANGE is specified and the mapping table name is provided.

T81REBLD
REBUILD job. This job rebuilds indexes.

T85REFR
REFRESH job. This job refreshes tables.

T89POSTI
Jobs to grant authorizations on tables and to reload accelerators.

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
One job is created per process. For example, all UNLOAD jobs are merged into one job for each process. 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.

The following jobs are created:

T10U0001
UNLOAD job.

T20DROP
DROP job.

T30CREAT
CREATE job.

T40STOP
STOP job.

T50ALTER
Alter job.

T60START
START job.

T70R0001
Reload job.

T89POSTI
Table GRANTs and reload accelerator job.

T71R0001
REORG job.

T72REBLD
REBUILD job.

T73CHECK
CHECK job.

T74IMC
IMAGECOPY job.

T80R0001
REORG job.

T81B0001
REBUILD job.

T87RUNST
RUNSTATS job.

Member prefix
Specify a prefix to use for the member name or names for the apply job. The default is APPLY.

If the number of steps for the apply job exceed the limit of 255, more than one job is generated. For example, if **Member prefix** is APPLY, the member names are APPLY001, APPLY002 and so on.

Member prefix does not apply if you specify No or `process` for **Generate one job**.

As work statement list
Specify whether to put the *apply changes* (those changes that are generated when you set **Generate apply jobs** to Yes) in a work statement list. You can either append to or replace the work statement list.

You can specify one of the following values for this option:

Yes
If the work statement list (provided in the **Worklist name** option) already exists, the **Specify Work Statement List Data Set (ADB2WLDA)** panel prompts you for the data set name. On this panel, you can also indicate whether you want to append to or replace the work statement list and whether you want to build a batch job to run the work statement list.

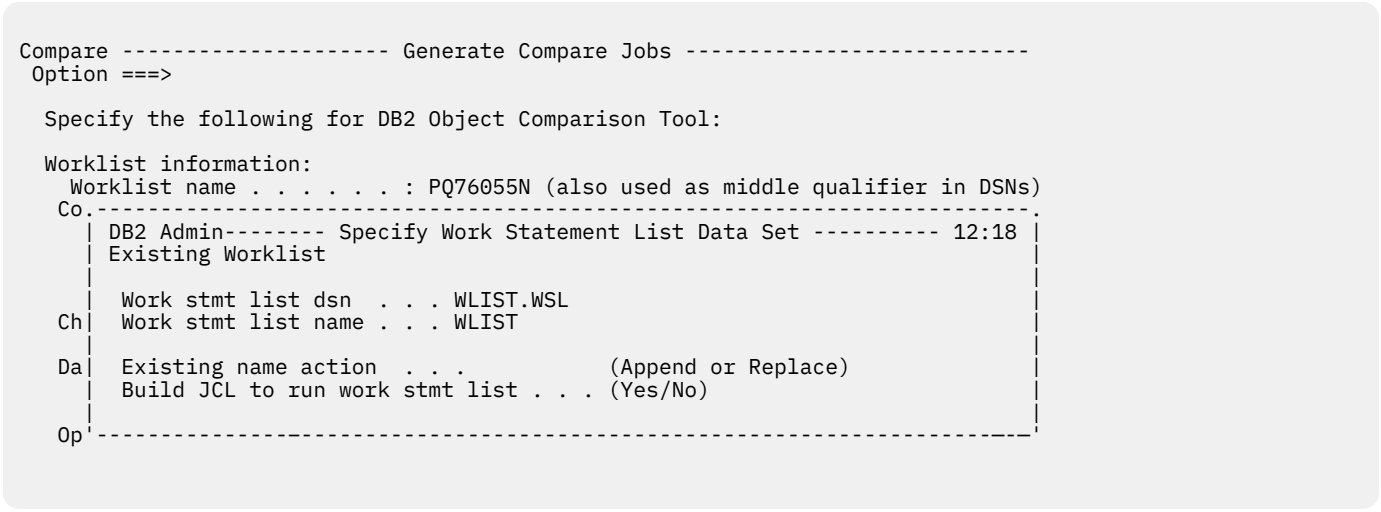

*Figure 40. **Specify Work Statement List Data Set (ADB2WLDA)** panel*

Restriction: The replace capability is not supported if you are using the MultiCompare function to compare more than one saved dialog. If you are using MultiCompare, work statement lists are automatically appended. To replace work statement lists for more than one dialog, you must run the comparisons individually.

If you are appending to an existing work statement list, the **Specify Work Statement List (ADB27WLD)** panel prompts you to specify a different middle qualifier to avoid reusing data sets:

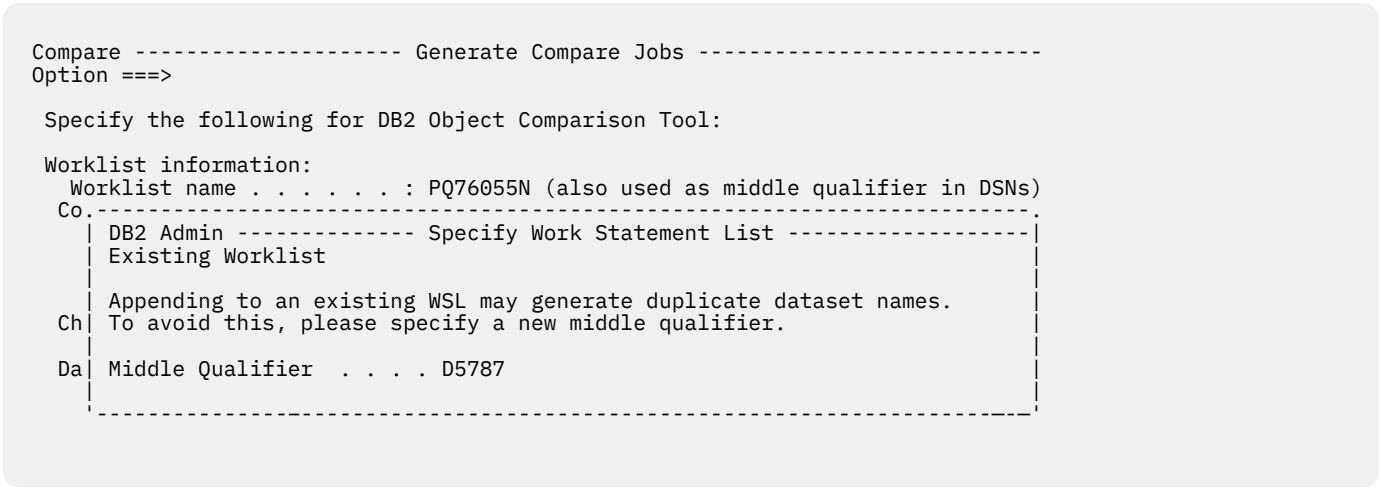

*Figure 41. **Specify Work Statement List (ADB27WLD)** panel*

If the work statement list (provided in the **Worklist name** option) does not exist, the **Specify Work Statement List Data Set (GOC5WL)** panel is displayed:

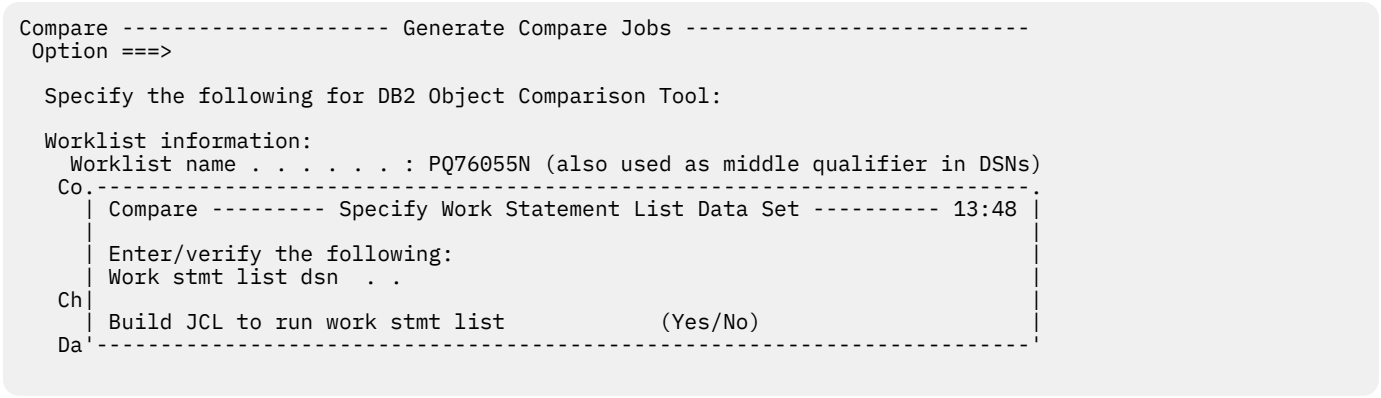

*Figure 42. **Specify Work Statement List Data Set (GOC5WL)** panel*

If you specify Yes for **Build JCL to run work stmt list**, the **Specify Job Parameters (ADB2W1R)** panel prompts you to specify the job library partitioned data set (PDS) and member prefix:

```
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 43. **Specify Job Parameters (ADB2W1R)** panel*

No
The apply jobs are generated in a separate data set. The **Specify Data Set Name for Apply Jobs (GOC5AJ)** panel prompts you for that data set name. 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 '-------------------------------------------------------------------------'
```

Embed IFF into WSL
Specify whether the WSL is to include the contents of the IFF file in an encoded format. An IFF file is produced if the change requires an unload operation. The benefit of embedding the IFF contents in the WSL is that you can transport the WSL to another system without having to separately transport the IFF file. The default value is NO.

Use customized util opts
Specify whether you want to use the options that you set for the COPY, CHECK DATA, MODIFY, REBUILD, REORG, RUNSTATS, UNLOAD, and LOAD utilities on the Db2 Administration Tool panels. If you specify Yes, utility jobs and work statement lists are generated based on the utility definitions that you specified. Otherwise, the default utility options are used.

Related information:

"UO - Change utility options" on page 98

Content of apply job(s)
Specify whether to generate only changes to database objects.

You can specify one of the following values for this option:

All
Generate all jobs and processes to reload the data.

DDL
Generate only DDL. Object Comparison Tool does not generate UNLOAD statements, LOAD statements or other utilities except for rebind and REORG operations that are needed to apply the pending definition changes and remove any restrictive states. These operations are necessary to allow the subsequent statements to be successful.

When DDL is specified, any data conversion errors are ignored and no conversion report is generated.

Unload method
Specify the method that you want to use to unload data.

You can specify one of the following values for this option:

Unload
Use the Db2 UNLOAD utility.

Parallel unload
Use the Db2 UNLOAD utility with parallel processing.

Parallel unload cannot be used in the following situations:

- A limit key change
- A change in number of partitions
- The use of an identity column in the partitioning key

If Db2 Object Comparison Tool determines that the operation is not eligible for a parallel unload, it uses `Unload` instead.

If the operation is eligible for a parallel unload, a template is used to allocate the unload data sets.

HPU
Use Db2 High Performance Unload for z/OS to unload the data. Db2 High Performance Unload for z/OS must be available.

If `Parallel unload` and HPU are not valid options for the current unload, Object Comparison Tool automatically uses the UNLOAD utility.

Generate templates
Specify whether you want the compare process to generate templates for data sets.

If you specify `Yes`, templates are generated for non-utility data sets with the definitions that you specified in Db2 Administration Tool.

If you specify `No`, the values for the **Prefix for data sets** and **Worklist name** options are used.

If the **Take an image copy** or **Run REORG** options are set to `Yes`, the utility templates are used.

Related information:

"TU - Specify TEMPLATE usage" on page 97
LISTDEFs and TEMPLATEs (IBM Db2 Administration Tool for z/OS 13.1.0)

Stop on conversion error
Specify whether you want the compare process to stop if a conversion error occurs.

If you specify Yes and a conversion errors occurs, the APPLY job is not generated, and an error message similar to the following message is displayed:

```
Compare table source(aaaaaa.bbbbbb) and target(aaaaaa.bbbbbb)
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.bbbbbb 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
```

Use DEFER YES
Specify whether to use DEFER YES clauses on any eligible CREATE INDEX statements.

If you specify Yes, DEFER YES is used for eligible indexes. However, any user-defined masks for the DEFER attribute take precedence over the value of this option.

If you specify DEFER YES = YES and Run REORG/REBUILD = A, REBUILD INDEX jobs will be generated. The DDL will still contain the DEFER YES clause, but that clause will be ignored and the indexes will be rebuilt.

Allow rotate parts
Specify whether to generate the ROTATE PARTITION statement or the ALTER PARTITION statement when the condition for a rotation is met.

Yes
Generate the ROTATE PARTITION statement. Data from the rotating partitions is unloaded before the rotation. You can discard this data or to load it back into the new partitions.

No
Generate the ALTER PARTITION statement and a REORG statement for the affected partitions. For the **Run REORG/REBUILD** option, specify either Mandatory or All Relevant to generate the REORG statements. Data from the rotating partitions is loaded back into the table so that you do not have to manually perform the reload. Logical and physical partitions are preserved.

If your table is not partitioned, specify Yes.

Retain GENERATED ALWAYS:

For ROWID
Specify whether to retain the GENERATED ALWAYS attribute for ROWID columns. Specify Yes or No.

For ROW CHANGE TIMESTAMP
Specify whether to retain the GENERATED ALWAYS attribute for ROW CHANGE TIMESTAMP columns. Specify Yes or No.

Retain START and RESTART values:

For sequence object:
Specify whether to retain START and RESTART values for the sequences. Specify Yes or No.

If you specify No and ignores are specified for the START or RESTART fields, the ignored fields are not changed. If you specify No and ignores are not specified for the START and RESTART fields, the values on the target are changed according to the source.

IDENTITY START value
Specify the value to use for the identity column when the table is re-created.

You can specify one of the following values for this option:

Original
Use the START value for the identity column from the Db2 catalog.

Computed
Compute the START value based upon the identity attributes of the column.

The computed value is based on the existing identity column attributes, such as MAXASSIGNED and the current cache size, at the time that the DDL is produced. Any changes made after the creation of the DDL, to either the identity values or to the data, are not reflected in the DDL and make the DDL obsolete. Object Comparison Tool does not locate an unassigned value from the existing data, and ultimately a new RESTART value might need to be provided based upon the underlying data and the application needs.

Mask ignored fields
Specify whether to apply masked values to ignored fields for newly added objects if the field has been masked and ignored.

If you specify No, the original values from the source are applied. No is the default.

This option is not applicable to ignore files that are provided in the **CM Register Options (ADB2CRO)** panel.

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 LOAD utility jobs that are generated by Db2 Object Comparison Tool to reload the data.

Recommendation: Specify Yes if LOAD uses ENFORCE NO.

Take an image copy
Specify whether to generate COPY utility jobs.

You can specify one of the following values for this option:

Reload
Generate a COPY utility job for all tables that are affected by the generated LOAD utility jobs to reload the data.

Alter
Generate a COPY utility job for all table spaces, tables, and indexes that are altered with generated ALTER statements.

Both
Generate a COPY utility job for all tables that are affected by the LOAD utility job and all altered table spaces, tables, and index objects.

None
Do not generate any COPY utility jobs.

Run REORG/REBUILD
Specify whether to generate REORG TABLESPACE utility jobs and REBUILD INDEX utility jobs, if needed. These jobs are run after applying the changes from an object comparison to make the target system operational.

You can specify one of the following values for this option:

Mandatory
Generate all REORG and REBUILD jobs that are needed to remove any REORG-pending and REBUILD-pending states and make the data available.

All relevant
Generate all REORG and REBUILD jobs that are needed to fully implement the changes. For example, changing PRIQTY is registered when the table space is altered, but the new value is not used until the table space is reorganized.

None
Do not generate any REORG and REBUILD jobs. This option is not valid if you specified No for **Allow rotate parts**.

Run RUNSTATS
Specify whether to generate RUNSTATS utility jobs.

You can specify one of the following values for this option:

Reload
Generate a RUNSTATS utility job for all tables that are affected by the generated LOAD utility jobs to reload the data.

Alter
Generate a RUNSTATS utility job for all table spaces, tables, and indexes that are altered with generated ALTER statements.

Both
Generates a RUNSTATS utility job for all tables that are affected by the LOAD utility job and all altered table spaces, tables, and index objects.

Minimum
Generate RUNSTATS utility jobs for the following conditions:

- If a table space is dropped and recreated, generate RUNSTATS statements for the tables and indexes.
- If a table is dropped and recreated, generate RUNSTATS statements for only the indexes and not the table.
- If an index is created, recreated, or has columns added, generate RUNSTATS statements for the index.
- If the index is created with DEFER YES and REBUILD is generated, the RUNSTATS operation is performed after the REBUILD operation.

None
Do not generate any RUNSTATS utility jobs.

Run REBIND
Specify whether to generate a job to rebind the plans and packages that are affected by the changes from an object comparison.

Mandatory
Generate a REBIND job for only those plans and packages that were invalidated by the changes.

All relevant
Generate a REBIND job for all plans and packages that were affected by the changes, including those plans and packages that were invalidated.

None
Do not generate a REBIND job.

REBIND options
Indicate whether you want to specify your own BIND options for any plans and packages that are rebound. To use this field, you must specify `M` (Mandatory) or `A` (All relevant) in the **Run REBIND** field.

YES
Allows you to specify BIND options on the subsequent **REBIND options (ADBPREBO)** panel. When this panel is displayed, specify any BIND options for dependent packages, and press Enter:

```
ADBPREBO  ----------------------- REBIND options -----------------------
Command ===>


Specify additional REBIND parameters to generate rebinds
for dependent packages.

  APREUSE . . . .                 (None, Warn, Error)
  EXPLAIN . . . .                 (Yes, No, Only)
  OWNER . . . . .               > (Owner of package)
    OWNERTYPE . .                 (Role, User)

  Additional options: ____________________________________________________
  ____________________________________________________________________________
  ____________________________________________________________________________
  ____________________________________________________________________________
  ____________________________________________________________________________
```

*Figure 44. **REBIND options (ADBPREBO)** panel*

Note: Any BIND options that you specify in the **Additional options** field are added to the REBIND statement as is; they are not validated.

NO
Any affected plans or packages are rebound with their existing BIND options (the options that were used during the previous bind or rebind operation).

Commands

You can enter the following commands on the command line:

BP - Change batch job parameters
Allows you to change the parameters for batch utility jobs, such as the job card and space parameters.

When you specify the BP command and press Enter, the **Batch Job Utility Parameters (ADB2UPA)** panel opens:

```
 DB2 Admin -------------- DD1A Batch Job Utility Parameters -------------- 11:02
Command ===>

Generate Job Card . ===>      (Yes/No)                  DB2 System: DD1A
 Job cards:                                            DB2 SQL ID: ADM001
  ===> //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)
```

Related information:

Batch job parameters for utility jobs (IBM Db2 Administration Tool for z/OS 13.1.0)

TU - Specify TEMPLATE usage

Allows you to modify templates for the data sets that are allocated and used by Db2 Object Comparison Tool.

When you specify the TU command and press Enter, the **Specify UTILITY TEMPLATE Usage (ADB25TU3)** panel opens:

```
ADB25TU3               DD1A Specify UTILITY TEMPLATE Usage              11:58
 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
```

From this panel, you can modify the templates for data sets. The default work data sets and descriptions are shown in the following table:

Table 6. Work data set descriptions

Template keyword	Default data set	Description
GOCALTR	*prefix.worklist*.DDL.ALTER	Primarily ALTER statements
GOCCREA	*prefix.worklist*.DDL.CREATE	Primarily CREATE statements
GOCDROP	*prefix.worklist*.DDL.DROP	Primarily DROP statements
GOCRBND	*prefix.worklist*.CMD.REBIND	REBIND control statements
GOCIFFN	*prefix.worklist*.IFF	Internal version file
GOCSHVR	*prefix.worklist*.SHRVARS	ISPF variables
GOCCHNG	*prefix.worklist*.CHANGES	Changes from compare

You can specify the following variables in templates:

- The following functional variables:

  &GOCPRE
  The prefix for data sets, which you specify on the **Generate Compare Jobs (GOC5)** panel.

  &GOCWLN
  The statement work list name, which you specify on the **Generate Compare Jobs (GOC5)** panel.
- The date and time variables that are supported for the Db2 TEMPLATE utility.
- &USERID

Related information:

Associating templates with data sets (IBM Db2 Administration Tool for z/OS 13.1.0)
Syntax and options of the TEMPLATE control statement (Db2 13 for z/OS)

UO - Change utility options
Allows you to specify options for Db2 utilities.

When you specify the UO command and press Enter, the **Change Utilities Options (ADB2UOPS)** panel opens where you can select the Db2 utility for which you want to change the options:

```
ADB2UOPS  ---------------- DD1A Change Utilities Options ---------------- 11:15

Select one of the following, then press Enter.

   C - Image copy
  KD - Check data
   M - Modify
   O - Reorg tablespace
  OI - Reorg index
  RB - Rebuild index
   R - Runstats tablespace
   U - Unload
   L - Load









Option
===>
```

When you press Enter, the **Specify Utility Options** panel for the utility opens and you can enter the options that you want. Press Enter to save your selections.

Restriction: Some utility options are not available for utility jobs that are built by Db2 Object Comparison Tool.

For Db2 Object Comparison Tool to generate utilities with the options that you specified, on the **Generate Compare Jobs (GOC5)** panel, you must set the **Use utility options** to Yes. The options that you select are retained and used for any subsequent jobs where **Use utility options** is set to Yes.

Related information:

Running Db2 utilities from Db2 Admin Tool (IBM Db2 Administration Tool for z/OS 13.1.0)

CO - Change options common to change functions

Allows you to review and change options that are common to change functions in Db2 Administration Tool and Db2 Object Comparison Tool.

When you specify the CO command and press Enter, the **Options for Change Functions (ADB2PCO)** panel opens:

```
ADB2PCO n              Options for Change Functions                    19:13
Command ===>
                                                        DB2 System: DD1A


Recreate accelerated tables  . . . . . YES (Yes/No. Default is Yes)
Restore replication of tables  . . . . YES (Yes/No. Default is Yes)
Reload accelerated tables  . . . . . . YES (Yes/No. Default is Yes)
Restore acceleration of tables . . . . YES (Yes/No. Default is Yes)
Remove deleted accelerated tables  . . YES (Yes/No. Default is Yes)

Load accelerated tables lock mode  . .             (Default is TABLESET)
Unload altered tables  . . . . . . . . NO   (Yes/No/Des. Default is YES)
Preserve all data  . . . . . . . . . . YES  (Yes/No. Default is YES)

Enable WSL authorization switching . . NO  (Yes/No. Default is No)
Object processing order  . . . . . . . H   (T - Object type, H - DB hierarchy.
                                             Default is H)
Statement validation exit name . . . . ________  (Name of EXEC used to validate
                                                  statements in WSL Validate)
Allow PBR2 to PBR changes . . . . . .  NO (Yes/No. Default is No)
DB2 release number . . . . . . . . . . ____ (Use VVRM format)
DB2 function level . . . . . . . . . . 502 (E.g. 100, 500, 501, 5nn)
Table GRANT processing order . . . . . C    (C - CREATE prefix for GRANT
                                             P - POSTUTIL prefix for GRANT
                                             Default is C )
```

Generating a compare batch job to make changes through Change Management

Registering changes in Change Management (CM) 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 during the customization of Db2 Admin Tool.

This procedure is a subtask of "5. Generating a compare batch job" on page 78. You must have identified the source and target objects and optionally any masks and ignores and have the **Generate Compare Jobs (GOC5)** panel displayed.

About this task

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

To make compare changes through CM:

1. On the **Generate Compare Jobs (GOC5)** panel, specify values for the compare job options as follows, and press Enter:

   Specify Yes for **Generate apply jobs**.

   Tip: Set **Save compare results** to YES if you want to analyze data about the comparison, ignore changes, or increase the efficiency of subsequent comparisons. The saved compare results contain information about objects that were part of the comparison, including detected differences, changes to make, and how those changes are to be implemented. You can save the compare results only for tables, indexes, global variables, and distinct data types.

   For more information about the other options on this panel, see "Compare job options" on page 80.

   If CM is mandatory for your SQL ID, the **CM Register Options (ADB2CRO)** panel is displayed.

   If CM is optional for your SQL ID, the **Change Management Prompt (ADB2CMRO)** panel is displayed:
2. If the **Change Management Prompt (ADB2CMRO)** panel is displayed, Specify Yes to process the compare change through Change Management, and press Enter.

   If you specify NO, the job is generated with the Apply step, and the changes are not registered in the CM database. Complete the steps in "5. Generating a compare batch job" on page 78 instead.

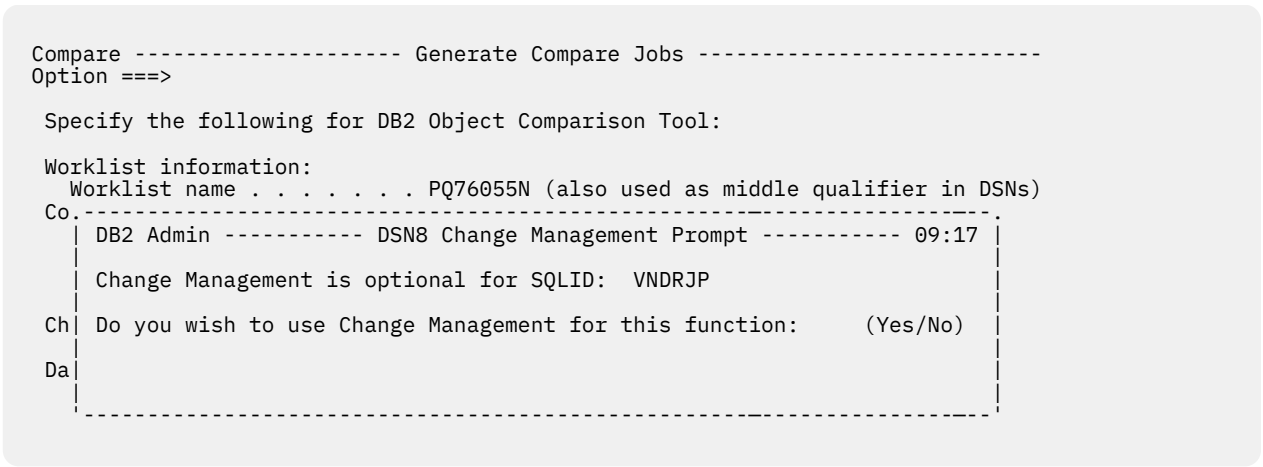

*Figure 45. **Change Management Prompt (ADB2CMRO)** panel*

3. On the **CM Register Options (ADB2CRO)** panel, specify the following information:
   - 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).
   - YES to register the change on multiple target locations. Optionally, specify a target profile.
   - Optionally, a comment for the change, whether to replace existing changes, an ignore for the change, and a mask for the change.

```
ADB2CRO n -------------------- CM - Register Options -------------------- 11:27
Command ===>

Commands: CONTINUE                                      DB2 System: DD1A
                                                        DB2 SQL ID: ADM001
Specify the following values to register a change:

Owner . . . . . . . . ADM001          > (Optional, Default is ADM001)
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 46. **CM Register Options (ADB2CRO)** panel*

4. Issue the NEXT command, and press Enter.

   The change is registered as a normal change.

5. Complete the requested input on any subsequent panels that are displayed.

   One or more of the following panels might be displayed depending on the compare job options that you selected. For more information about one of these panels, see the related job option description.

Panel	Option on Generate Compare Jobs (GOC5) panel
Save Compare Results (ADB2C22) panel	“Save compare results” on page 84
Specify Compare Reporting Options (GOC5RO) panel	“Change reporting options” on page 83
Specify Data Set Name for Apply Jobs (GOC5AJ) panel	“Generate apply jobs” on page 86
Specify Work Statement List Data Set (ADB2WLDA) panel	“As work statement list” on page 89
Specify Work Statement List (ADB27WLD) panel	“As work statement list” on page 89
Specify Work Statement List Data Set (GOC5WL) panel	“As work statement list” on page 89
Specify Job Parameters (ADB2W1R) panel	“As work statement list” on page 89
REBIND options (ADBPREBO) panel	“REBIND options” on page 95
DB2 Object Compare Warning (GOCGCMPW) panel	“Generate apply jobs” on page 86

6. On the **Specify Register Mode (GOC5RM)** panel specify one of the following actions 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 pending changes exist.

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.

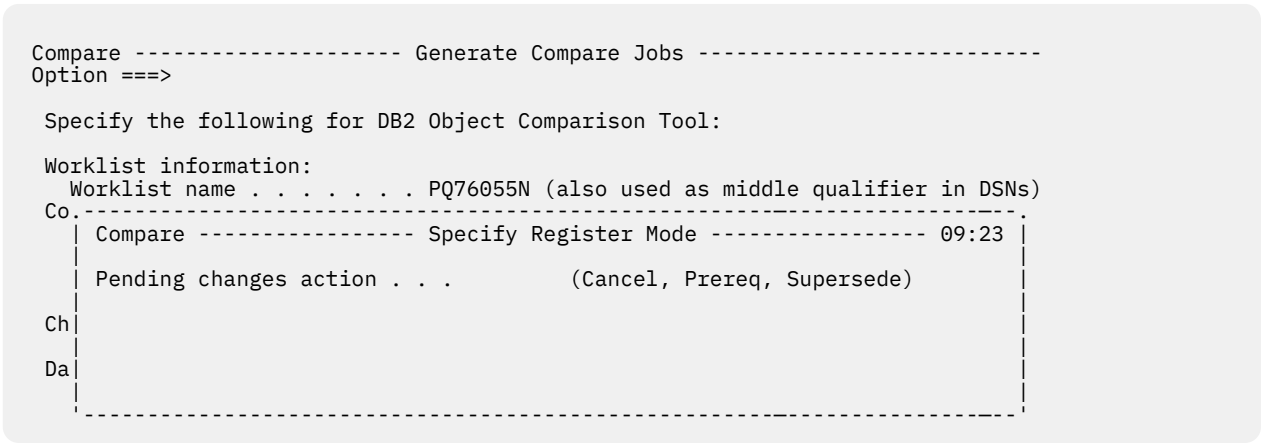

*Figure 47. **Specify Register Mode (GOC5RM)** panel*

7. Edit the generated JCL job as needed and submit it to run the comparison.

   For information about the Object Comparison Tool parameters in the generated JCL job, see "Parameters in the generated compare batch job" on page 106.

Results

The change to apply the compare changes is registered. After the compare batch job is run, you can use Db2 Admin Tool to analyze and run the change. These actions apply the changes from the comparison.

Related information

Analyzing a change (IBM Db2 Administration Tool for z/OS 13.1.0)

Running a change (IBM Db2 Administration Tool for z/OS 13.1.0)

Generating a compare batch job for a multi-target import

A *multi-target import* is the process of importing changes from a compare job to objects on multiple target environments. When you generate a batch job for this import, register the change with Change Management (CM) so that you can later import the change to objects on multiple target environments.

Before you begin

This procedure is a subtask of "5. Generating a compare batch job" on page 78. You must have identified the source and target objects and optionally any masks and ignores and have the **Generate Compare Jobs (GOC5)** panel displayed.

About this task

When you import changes from a compare job to objects on multiple target environments, you can specify masks for the compare job on one or more of the following panels:

Specify Compare Masks (GOC3) panel
Specify masking on this panel if your compare source and target object use different naming conventions.

CM Register Options (ADB2CRO) panel
Specify masking on this panel if your multi-target change uses different naming conventions than your compare target.

CM - Update Associated Target panel (ADBPCMTU) panel
Specify masking on this panel if your multiple target systems use different naming conventions than your multi-target change.

The masking that you specify on one panel does not override the masking that you specify on another panel. All specified masks are applied.

Procedure

To generate a compare batch job for a multi-target import:

1. On the **Generate Compare Jobs (GOC5)** panel, specify values for the compare job options as follows, and press Enter:

   Specify No for **Generate Online**.

   Tip: Set **Save compare results** to YES if you want to analyze data about the comparison, ignore changes, or increase the efficiency of subsequent comparisons. The saved compare results contain information about objects that were part of the comparison, including detected differences, changes to make, and how those changes are to be implemented. You can save the compare results only for tables, indexes, global variables, and distinct data types.

   For more information about the other options on this panel, see “Compare job options” on page 80.
2. If the **Change Management Prompt (ADB2CMRO)** panel is displayed (because CM is enabled and optional for your ID), specify YES, and press Enter.
3. On the **CM Register Options (ADB2CRO)** panel, specify a name for the change and set **Multi-target change** to YES. You can also optionally specify a mask.

```
ADB2CRO n -------------------- CM - Register Options -------------------- 16:25
 Command ===>

 Commands: CONTINUE                                      DB2 System: DD1A
                                                         DB2 SQL ID: ADM001
 Specify the following values to register a change:

 Owner . . . . . . . . ADM001          >  (Optional, Default is ADM001)
 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  . . . . . . . .           >                                    >
```

4. Issue the NEXT command, and press Enter.
5. If the **Insert a Target (ADBPC911)** panel is displayed, complete the fields to add a target, and press Enter.

   For help on defining targets for Change Management, see Setting up the targets (IBM Db2 Administration Tool for z/OS 13.1.0).
6. On the **Associate Targets (ADBPCMT)** panel, use the line commands to add and edit targets as needed.

```
DB2 Admin --------------- CM - Associate Targets ------------- Row 1 to 1 of 1
 Command ===>                                                  Scroll ===> CSR

 Details for multi-target change: PEDRO.TEST1             DB2 System: DD1A
                                                          DB2 SQL ID: ADM001
 Commands: NEXT
 Line commands:
  U - Update  D - Delete  AT - Add targets  AG - Add targets from group
  I - Interpret  ? - Show all line
commands


     Target                 Change    Change
 Sel Name     DB2 Location  Owner     name                    Status
     *        *             *         *                       *
 --- -------> ------------> --------- ---------------------- -----------------
     PSVTEST  DBAD                                           NEW
 ****************************** END OF DB2 DATA ******************************
```

7. Issue the NEXT command, and press Enter.
8. Complete the requested input on any subsequent panels that are displayed.

   One or more of the following panels might be displayed depending on the compare job options that you selected. For more information about one of these panels, see the related job option description.

Panel	Option on Generate Compare Jobs (GOC5) panel
Save Compare Results (ADB2C22) panel	“Save compare results” on page 84
Specify Compare Reporting Options (GOC5RO) panel	“Change reporting options” on page 83
Specify Data Set Name for Apply Jobs (GOC5AJ) panel	“Generate apply jobs” on page 86
Specify Work Statement List Data Set (ADB2WLDA) panel	“As work statement list” on page 89
Specify Work Statement List (ADB27WLD) panel	“As work statement list” on page 89
Specify Work Statement List Data Set (GOC5WL) panel	“As work statement list” on page 89
Specify Job Parameters (ADB2W1R) panel	“As work statement list” on page 89
REBIND options (ADBPREBO) panel	“REBIND options” on page 95
DB2 Object Compare Warning (GOCGCMPW) panel	“Generate apply jobs” on page 86

9. On the **Specify Register Mode (GOC5RM)** panel specify one of the following actions 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 pending changes exist.

   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.

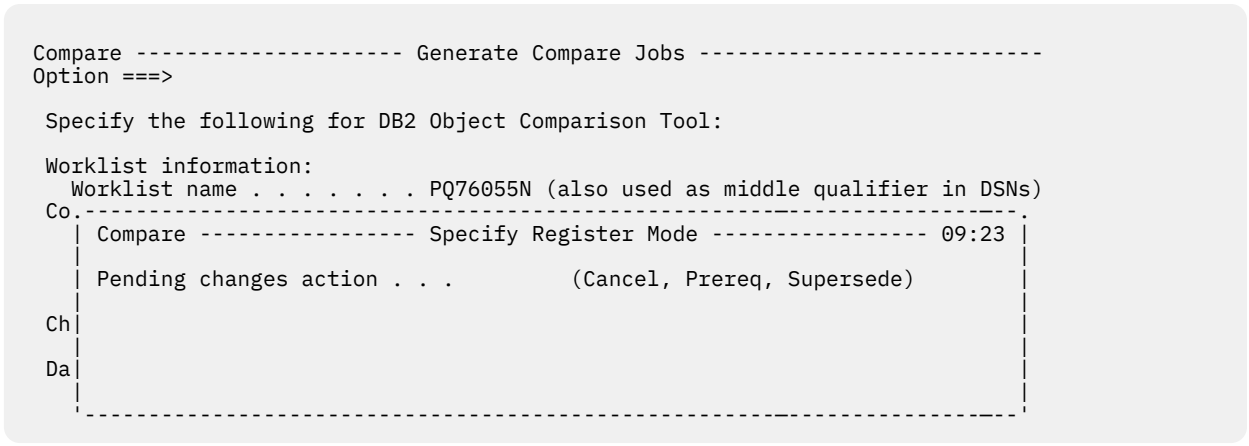

*Figure 48. **Specify Register Mode (GOC5RM)** panel*

10. Edit the generated JCL job as needed and submit it to run the comparison.

    For information about the Object Comparison Tool parameters in the generated JCL job, see "Parameters in the generated compare batch job" on page 106.

Updating an associated target

When creating multi-target changes, you might need to update the details for one of your target environments.

Procedure

To update a target:

1. On the **DB2 Object Comparison Tool Menu (GOCMENU)** panel, specify option 5, and press Enter.
2. On the **Generate Compare Jobs (GOC5)** panel, specify values for the compare job options, and press Enter.
3. If the **Change Management Prompt (ADB2CMRO)** panel is displayed (because CM is enabled and optional for your ID), specify YES, and press Enter.
4. On the **CM Register Options (ADB2CRO)** panel, set **Multi-target change** to YES.
5. Issue the NEXT command, and press Enter.

   If the Multi-target change field is set to YES, panel ADBPCMT is displayed. If NO is specified, the compare job is built immediately.
6. Specify the U line command next to the target you wish to update.

   The **CM - Update Associated Target panel (ADBPCMTU)** 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 49. **CM - Update Associated Target panel (ADBPCMTU)** panel*

7. 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.
8. 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.

Parameters in the generated compare batch job

When you generate a compare batch job, the resulting JCL includes a number of Object Comparison Tool parameters. Those parameters correspond to the following panel options:

Parameter in the JCL job	Panel	Corresponding option
ACCLOCKMODE	**Options for Change Functions (ADB2PCO)** panel	**Load accelerated tables lock mode** (See “ CO - Change options common to change functions” on page 99.)
ALTPART	**Generate Compare Jobs (GOC5)** panel	Allow rotate parts
APPCONT	**Generate Compare Jobs (GOC5)** panel	“Content of apply job(s)” on page 92
AUTHSQL	**ALTER - Build Analyze and Apply Job (ADBPALT)** panel	**Authorization Switch ID**
AUTHSW	**Generate Compare Jobs (GOC5)** panel	“Enable auth-switching” on page 82
CMDDL	None	None CMDDL does not correspond to a panel option. CMDDL specifies whether to comment out the ADMIN ALTER IMPLICIT statements.

Parameter in the JCL job	Panel	Corresponding option
CMDELTA	**Generate Compare Jobs (GOC5)** panel	Generate Apply Job = Change
CMPRACT	**Generate Compare Jobs (GOC5)** panel	“Save compare results” on page 84
CMPRADEL	**Save Compare Results (ADB2C22)** panel	**Eligible for auto-delete**
CMPRCOMM	**Save Compare Results (ADB2C22)** panel	**Comment**
CMPRNAME	**Save Compare Results (ADB2C22)** panel	**Name**
CMPROWN	**Save Compare Results (ADB2C22)** panel	**Owner**
DACVE	**Generate Compare Jobs (GOC5)** panel	“Stop on conversion error ” on page 92
DISOPTRE	**Generate Compare Jobs (GOC5)** panel	“Disable REORG optimization” on page 83
DROP_FKS_NOT_IN_SOURCE	**Generate Compare Jobs (GOC5)** panel	“Drop FKs not in source” on page 81
DROPEXOBJ	**Generate Compare Jobs (GOC5)** panel	“Allow implicit drop of excluded objects ” on page 82
ENACCAT	**Options for Change Functions (ADB2PCO)** panel	**Restore acceleration of tables** (See “ CO - Change options common to change functions” on page 99.)
ENREPAT	**Options for Change Functions (ADB2PCO)** panel	**Restore replication of tables** (See “ CO - Change options common to change functions” on page 99.)
GRANTORD	**Options for Change Functions (ADB2PCO)** panel	**Table GRANT processing order** (See “ CO - Change options common to change functions” on page 99.)
GRTSQLID	**Generate Compare Jobs (GOC5)** panel	“Object Grantor” on page 82
ICSPECNAME	**Specify Compare Ignores (GOC4)** panel	**Ignore Changes Specification: Name** (See “Modifying ignore changes specifications” on page 123.)
ICSPECOWN	**Specify Compare Ignores (GOC4)** panel	**Ignore Changes Specification: Owner** (See “Modifying ignore changes specifications” on page 123.)
IDENTSVL	**Generate Compare Jobs (GOC5)** panel	“IDENTITY START value” on page 93
KEEPCOL	**Generate Compare Jobs (GOC5)** panel	“Suppress DROP of columns” on page 81

Parameter in the JCL job	Panel	Corresponding option
KEEPTGT	**Generate Compare Jobs (GOC5)** panel	"Suppress DROP of objects" on page 81
MASKIGN	**Generate Compare Jobs (GOC5)** panel	"Mask ignored fields" on page 94
NONEWCOL	**Generate Compare Jobs (GOC5)** panel	"Suppress adding columns" on page 81
PBR2TOPBR	**Options for Change Functions (ADB2PCO)** panel	**Allow PBR2 to PBR changes** (See " CO - Change options common to change functions" on page 99.)
PROCORD	**Options for Change Functions (ADB2PCO)** panel	**Object Processing order** (See " CO - Change options common to change functions" on page 99.)
REBIND_APLJOB	None	None REBIND_APLJOB does not correspond to a panel option. REBIND_APPLJOB specifies whether the REBIND parameters need a trailing hyphen in the apply job format in the resulting JCL when not using a WSL for the change.
REBIND_APREUSE	**REBIND options (ADBPREBO)** panel	**APREUSE** [See Figure 44 on page 96.]
REBIND_EXPLAIN	**REBIND options (ADBPREBO)** panel	**EXPLAIN** [See Figure 44 on page 96.]
REBIND_OWNER	**REBIND options (ADBPREBO)** panel	**OWNER** [See Figure 44 on page 96.]
REBIND_OWNERTYPE	**REBIND options (ADBPREBO)** panel	**OWNERTYPE** [See Figure 44 on page 96.]
REBIND_ADD_OPTS	**REBIND options (ADBPREBO)** panel	**Additional options** [See Figure 44 on page 96.]
REBINDA	**Generate Compare Jobs (GOC5)** panel	Run REBIND = All relevant
REBINDM	**Generate Compare Jobs (GOC5)** panel	Run REBIND = Mandatory
REBINDN	**Generate Compare Jobs (GOC5)** panel	Run REBIND = None
RECOVER	**Generate Analyze Job (ADB2C11A)** panel	**Data to recover** (This parameter is for Change Management use only. See Analyzing a change (IBM Db2 Administration Tool for z/OS 13.1.0).)
RECREAT	**Options for Change Functions (ADB2PCO)** panel	**Recreate accelerated tables** (See " CO - Change options common to change functions" on page 99.)
RELOADAT	**Options for Change Functions (ADB2PCO)** panel	**Reload accelerated tables** (See " CO - Change options common to change functions" on page 99.)

Parameter in the JCL job	Panel	Corresponding option
REMOVEAT	**Options for Change Functions (ADB2PCO)** panel	**Remove deleted accelerated tables** (See " CO - Change options common to change functions" on page 99.)
REORGA	**Generate Compare Jobs (GOC5)** panel	Run REORG/REBUILD = All relevant
REORGM	**Generate Compare Jobs (GOC5)** panel	Run REORG/REBUILD = Mandatory
REPALL	**Specify Compare Reporting Options (GOC5RO)** panel	All possible reporting options are set to yes, except REPCHG. See "Reporting options:" on page 83 on **Generate Compare Jobs (GOC5)** panel
REPCHG	**Specify Compare Reporting Options (GOC5RO)** panel	"Only changed objects" on page 83
REPCONV	**Specify Compare Reporting Options (GOC5RO)** panel	"Conversion report" on page 84
REPCOUNT	**Specify Compare Reporting Options (GOC5RO)** panel	"Object count report" on page 84
REPIGALL	**Specify Compare Reporting Options (GOC5RO)** panel	All Ignore fields reporting options are set to yes.
REPIGOSI	**Specify Compare Reporting Options (GOC5RO)** panel	Ignore fields: Object Specific
REPIGSYS	**Specify Compare Reporting Options (GOC5RO)** panel	Ignore fields: System generated
REPIGUSR	**Specify Compare Reporting Options (GOC5RO)** panel	Ignore fields: User specified
REPMASK	**Specify Compare Reporting Options (GOC5RO)** panel	"Translation masks" on page 83
REPSUM	**Specify Compare Reporting Options (GOC5RO)** panel	"Summary report" on page 83
RIDALWYS	**Generate Compare Jobs (GOC5)** panel	Retain GENERATED ALWAYS: For ROWID
RPTEXOBJS	**CM - Manage Exclude Specifications (ADBPC7)** panel	Excluded objects (See Creating and managing exclude specifications (IBM Db2 Administration Tool for z/OS 13.1.0).)
RPTEXSPEC	**CM - Manage Exclude Specifications (ADBPC7)** panel	Exclude specifications (See Creating and managing exclude specifications (IBM Db2 Administration Tool for z/OS 13.1.0).)

Parameter in the JCL job	Panel	Corresponding option
SCOPEWARN	**Generate Compare Jobs (GOC5)** panel	"Scope Warning Messages" on page 83
SEQSRVL	**Generate Compare Jobs (GOC5)** panel	Retain START and RESTART values for sequence object
SRCIESPECNAME	**Specify Compare Source (GOC1)** panel	**Exclude Specification: Name** (See "1. Specifying source objects" on page 48.)
SRCIESPECOWN	**Specify Compare Source (GOC1)** panel	**Exclude Specification: Owner** (See "1. Specifying source objects" on page 48.)
TGTIESPECNAME	**Specify Compare Target (GOC1)** panel	**Exclude Specification: Name** (See "2. Specifying target objects" on page 60.)
TGTIESPECOWN	**Specify Compare Target (GOC1)** panel	**Exclude Specification: Owner** (See "2. Specifying target objects" on page 60.)
TMSALWYS	**Generate Compare Jobs (GOC5)** panel	Retain GENERATED ALWAYS: For ROW CHANGE
UNLDALTB	**Options for Change Functions (ADB2PCO)** panel	**Unload Altered tables** (See " CO - Change options common to change functions" on page 99.)

Saving dialogs

In Object Comparison Tool, you can save the current compare batch job selections, including the options on the **Generate Compare Jobs (GOC5)** panel and its subordinate panels, for later retrieval. This set of user selections is called a *dialog*. You can later restore this dialog or use this dialog to compare multiple sources and targets.

Procedure

To save a dialog:

1. On the **DB2 Object Comparison Tool Menu (GOCMENU)** panel, specify option **S**, and press Enter.
2. On the **Save Dialog (ADB2SDS)** panel, specify the following information, and press Enter to save the dialog:

```
ADB2SDS ---------------------- Save Dialog ---------------------- 10:35

Enter/verify the following:
  Prefix      ===> NBRON.ADMIN.SAVEDLGS.APAR.B37
  Name        ===>
  Description ===>
```

Prefix
Specify a unique qualified name 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.

Name
Specify a name 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.

Description
Optionally, enter a description of the dialog.

What to do next

If you later want to retrieve this dialog, complete the steps in "Managing and restoring dialogs" on page 111.

Related tasks

"Comparing multiple sources and targets" on page 131
You can compare sources and targets from one or more saved dialogs. For example, you can compare multiple saved targets to one saved source. This functionality is called *MultiCompare*.

Managing and restoring dialogs

You can restore, rename, and delete previously saved dialogs.

Procedure

To manage and restore dialogs:

1. On the **DB2 Object Comparison Tool Menu (GOCMENU)** panel, specify option **M**, and press Enter.
2. On the **Saved Dialogs (ADB2SD)** panel, specify the prefix for the saved dialog data set, and press Enter.

```
Compare -------------------------- Saved Dialogs ----------------------- 14:05
Option ===>

Enter the prefix for saved dialog data sets:
  Prefix ===> NBRON.ADMIN.SAVEDLGS.APAR.B37
```

*Figure 50. **Saved Dialogs (ADB2SD)** panel*

The **Manage/Restore Dialogs (ADB2SDM)** panel lists all the saved dialogs in the specified dialog data set:

```
ADB2SDM n ---------------------- Manage/Restore Dialogs ---------- Row 1 of 2
Command ===>                                                Scroll ===> PAGE

Line commands:  D - Delete  R - Rename  S - Select  ? - Show all line
commands

S Name     Description                          Created    Id
  *        *                                    *          *
- -------- ------------------------------------ ---------- --------
  TEST01   SOURCE IS DDL W/ADD.COLUMN           2003/07/09 NBRON
  PREVTEST DDL&DB2 W/STOGROUP                   2003/09/09 NBRON
```

*Figure 51. **Manage/Restore Dialogs (ADB2SDM)** panel*

3. Issue one of the following line commands next to the appropriate dialog, and press Enter:

   S

   Restore the dialog. The selections for the corresponding dialog are restored, and you can immediately run the batch job.

   D

   Delete the dialog.

   R

   Rename the dialog. On the subsequent **Member Rename (ADB2SDR)** panel, enter the new dialog name, and press Enter.

Chapter 5. Checking the compare report

After you run a comparison, check the compare report to see the differences between the source and target objects.

About this task

If you want to change the target objects to match the source objects, first ensure that the compare report contains only the changes that you want to apply. You might need to create additional masks and ignore fields and repeat the comparison process until the report contains only the changes that you want.

Related tasks

“3. Specifying compare masks” on page 63
“4. Specifying ignores” on page 73

Managing saved compare results

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

Procedure

To manage saved compare results:

1. On the **DB2 Object Comparison Tool Menu (GOCMENU)** panel, specify option MR, and press Enter.
2. On the **Manage Saved Compare Results (ADBPMCR)** panel, specify criteria for the saved compare results that you want to view or modify, and press Enter:

```
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 . . . .
```

3. On the **Manage Saved Compare Results (ADBPCR)** panel, use the line commands to view more detail for, 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  ? - Show all line
commands



Sel Owner    Name                           Eligible for Comment
                                            auto-delete
--- -------- ------------------------------> -----------  --------------------->
S   OWN1     NEW1                           2012-12-31   My first compare result
    OWN2     NEW2
    OWN3     NEW3
    OWN4     NEW4
    OWN5     NEW5
    OWN6     NEW6
    OWN7     NEW7
```

*Figure 52. **Manage Saved Compare Results (ADBPCR)** panel*

For example, consider the following two commonly used line commands:

S

Use this line command to view the results of a particular compare operation.

The **Compare Results (ADBPCRS)** panel shows the summary of that compare operation:

```
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
  ? - Show all line commands

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

*Figure 53. **Compare Results (ADBPCRS)** panel*

For information about this panel, see "Compare Results (ADBPCRS) panel" on page 115.

RPT

Use this line command to view a report for a particular compare operation.

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 54. **Compare Report (ADBPCRR)** panel*

Compare Results (ADBPCRS) panel

The **Compare Results (ADBPCRS)** panel shows a summary of the selected compare operation. Open this panel by specifying line command S on the **Manage Saved Compare Results (ADBPCR)** panel.

```
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
  ? - Show all line commands

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

*Figure 55. **Compare Results (ADBPCRS)** panel*

This panel includes the following columns:

Sel
An input field for entering a line command.

Object
The type of object that was compared. This type can be one of the following values:

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
The referential constraint name if the change updates a referential constraint.

Compare Action
The type of change to the object.

Implicit
An indication of whether the change includes an implicit drop or an implicit drop and recreate.

Use the following commands to view additional information:

SHOWSOURCE
Displays the source object that is associated with the target in the **Additional Information** column.

VERSION
Displays the following fields:

Version
Shows the version number of native stored procedures and PL/SQL functions. This column is displayed in place of the **Compare Action** column.

Active version
Indicates an active version of specific native stored procedures and PL/SQL functions. This column is displayed in place of the **I** column.

Excluding objects from the compare process

To be more selective about which objects are included in a comparison, you can specify that certain objects are to be excluded from the compare process. To do so, you must define an *exclude specification*.

About this task

An *exclude specification* is a defined list of objects that Object Comparison Tool is to exclude from the source, target, or both. The exclude specification can include authorizations. You can exclude authorizations independent from the object with which they are associated.

Restriction: You cannot manage exclude specifications when you are analyzing or running a change in Change Management.

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 indexes and views, are also excluded. Any objects to be excluded must be explicitly included in an exclude specification. The exceptions are history tables, schemas, and archive tables:

- If a temporal table is excluded, its history table is also excluded. If the history table is specified in an exclude specification, both the temporal and history table are excluded.
- If a schema is excluded, all UDFs, UDTs, procedures, triggers, and sequences that are part of the schema are excluded.
- When an archive-enabled table is excluded, its corresponding archive table is also excluded. Similarly, if an archive table is excluded, its corresponding archive-enabled table is also excluded.

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

The following scenarios illustrate some of the uses of exclude specifications:

- 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, however, you exclude authorizations from the source, no message is issued.
- When comparing objects, if an object exists in the target only, the resulting action is to drop the object from the target. However, if the object is excluded, 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.
- When you use the Db2 Administration Tool GEN function, any authorizations in the exclude specification are excluded from DDL generation.

Excluded objects can still be dropped implicitly as a result of a comparison, depending on the setting of the **Allow implicit drop of excluded objects** field on the **Generate Compare Jobs (GOC5)** panel. An *implicit drop* occurs when the action of dropping an object results in the drop of a dependent object. For example, if a comparison results in a table space being dropped at the target, the table, index, and other objects dependent on the table space should also be dropped. If **Allow implicit drop of excluded objects** is set to YES, excluded objects can be dropped as needed and are re-created according to the target definition. If this option is set to NO, if an excluded object needs to be dropped (perhaps because its parent object was dropped or because its dependent object was dropped), Object Comparison Tool stops processing the compare, and a severe message is issued.

Procedure

To exclude objects from a comparison:

1. Create an exclude specification by using one of the following methods:
   - Define one in Change Management (CM). See Creating and managing exclude specifications (IBM Db2 Administration Tool for z/OS 13.1.0).
   - Create one based on saved compare results. See “Creating exclude specifications from saved compare results” on page 120
   - Create one during the compare process. See “Creating or editing exclude specifications during a comparison” on page 119.

   When adding and editing objects in an exclude specification on the **CM - Exclude Objects (ADBPC7L)** panel, use the following guidance:

   - You can use the wildcard character (*) when you are specifying object names.
   - 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.
   - In the **T** column, specify one of the following two-character codes for the object type or authorization type:
     - AL - Alias
     - CA - Column authorizations
     - 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
     - TA - Table authorization
     - TB - Table
     - TG - Trigger
     - TS - Table space
     - VA - View authorization
     - VW - View

2. Reference the exclude specification when specifying the source or target. See “1. Specifying source objects” on page 48 or “2. Specifying target objects” on page 60 or both.

Creating or editing exclude specifications during a comparison

You can create exclude specifications to omit objects from the compare process. If an exclude specification is not already defined, you can create one when selecting the source or target objects. You can also edit an existing exclude specification.

Before you begin

This procedure assumes that you are in the process of specifying source or target objects and the **Specify Compare Source (GOC1)** panel or the **Specify Compare Target (GOC1)** panel is displayed.

About this task

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 the specified object.

Procedure

To create or edit exclude specifications during a comparison

1. On the **Specify Compare Source (GOC1)** panel or the **Specify Compare Target (GOC1)** panel, complete the fields under **Exclude Specifications:** as follows:

   Name
   Specify the name for the exclude specification. If you want to edit an existing exclude specification, specify the existing name. If you want to create a new exclude specification, specify a unique name; the specification will be created.

   Owner
   Optionally specify the specification owner. If this field is left blank, your authorization ID is used as the owner.

   Edit Objects
   Specify YES.

2. Ensure that the option you want is specified in the **Option** field (according to the task you are completing: “1. Specifying source objects” on page 48 or “2. Specifying target objects” on page 60), and press Enter.
3. Complete the one of the following procedures:

Option	Description
To create a new exclude specification	a. On the **Insert Exclude Specification (ADB2C22)** panel, optionally specify a comment and an **Eligible for auto-delete** value, and press Enter. The following message confirms that the new specification was added: `INSERT stmt executed` b. Press exit (PF3). c. On the **CM - Exclude Objects (ADBPC7L)** panel, specify the objects that you want to include in the exclude specification. Type object names and other information, and use line commands to edit the list of objects. For guidance, see step “1” on page 118 in “Excluding objects from the compare process” on page 117. d. Press exit (PF3).
To edit an existing exclude specification	a. On the **CM - Exclude Objects (ADBPC7L)** panel, edit the list of objects that you want to include in the exclude specification. Type over the object names and other information, and use line commands to modify the list. For

Option	Description
	guidance, see step “1” on page 118 in “Excluding objects from the compare process” on page 117. b. Press exit (PF3).

What to do next

Return to “1. Specifying source objects” on page 48 or “2. Specifying target objects” on page 60.

Creating exclude specifications from saved compare results

Your saved compare results might include objects that you do not want to include in future comparisons. You can select these objects from the saved compare results and add them to an exclude specification. You can exclude them from the source, target, or both.

About this task

Restriction: Compare results are saved for only the following objects:

- tables
- indexes
- global variables
- distinct data types

Therefore, this method of creating exclude specifications (from the saved compare results) applies to only these objects.

Procedure

To create exclude specifications from saved compare results:

1. On the **DB2 Object Comparison Tool Menu (GOCMENU)** panel, specify option MR, and press Enter.
2. On the **Manage Saved Compare Results (ADBPMCR)** panel, specify any filtering criteria for the saved compare results that you want displayed, and press Enter.
3. On the **Manage Saved Compare Results (ADBPCR)** panel, specify the S line command next to the result that you want to select, and press Enter.
4. On the **Compare Results (ADBPCRS)** panel, specify one of the following line commands next to the object that you want to exclude and complete the resulting panel:

Option	Description
EX	**Exclude** On the resulting **Create Exclude Specification (ADBPCEX)** panel, specify the requested information about for the exclude specification, and press Enter:

Option	Description

	``
	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)
	``

EXS	**Exclude from source**
	The **CM - Exclude Objects (ADBPC7L)** panel displays a list of source objects that are generated from the compare run. Use the line commands to edit the list of objects that you want to exclude:
	``
	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 ? - Show all line
	commands
	                                  Column/ Auth
	Sel T Qual Name Constrnt Grantee Level Message
	    * * * * * *
	--- -- -------> ----------------> -------> -------> -------> --------------->
	    TB VNDRG TB23367
	    SA DB23367 TS23367B
	    SA DB23367 TS23367B
	``

EXT	**Exclude from target** The **CM - Exclude Objects (ADBPC7L)** panel displays a list of target objects that are generated from the compare run. Use the line commands to edit the list of objects that you want to exclude.

5. Exit back to the **DB2 Object Comparison Tool Menu (GOCMENU)** panel.

Ignoring changes

When you compare objects, the report might include object changes that you do not want. You can designate those object changes as *ignore changes*, or changes to ignore in subsequent compare processing. Specifying such changes is called an *ignore changes specification* and requires saved compare results.Within the saved compare results, you select the object changes that you want to ignore. The selected object changes will be reported as part of the compare process, but no SQL statements will be generated for the changes.

About this task

Only changes that report differences between a source object and a target object can be ignored. Added and dropped objects can be excluded from compare processing but not ignored.

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

You must specify each object change that you want ignored. Related object changes are not automatically ignored.

Tip: Use caution when selecting object changes to be ignored. 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 a change to the number of table space partitions is ignored, other object changes related to partitioning also need to be ignored. In this case, changes to the number of partitions in the table and adding or deleting a limit key must also be ignored. LOB columns and LOB objects are another example of an interdependency. If adding a LOB column to a table is ignored, adding an explicit auxiliary table for the column must also be ignored, and its explicit LOB table space and index be excluded from compare.

Procedure

To ignore changes:

1. Create an ignore changes specification.
2. Update the specification as needed during the compare process. See "Modifying ignore changes specifications" on page 123.

Creating ignore changes specifications

An *ignore changes specification* designates the changes that you want ignored in subsequent compare processing.

Before you begin

You must have saved compare results from which you want to select object changes to ignore.

Procedure

To create an ignore change specification:

1. On the **DB2 Object Comparison Tool Menu (GOCMENU)** panel, specify option MR (Manage saved compare results), and press Enter.
2. On the **Manage Saved Compare Results (ADBPMCR)** panel, specify criteria to find the desired saved compare result, and press Enter.
3. On the **Manage Saved Compare Results (ADBPCR)** panel, issue the RPT command., and press Enter.
4. On the **Compare Report (ADBPCRR)** panel, issue the I line command next to the compare changes that you want to ignore.

```
ADBPCRR  ------------------  DD1A 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 56. Compare Report panel (ADBPCRR)

When you press Enter, the status column for the change is updated to `I`.

5. Issue the `CONTINUE` command, and press Enter.
6. On the **Create Ignore Specification (ADB2C22)** panel, type an owner name and specification name.
   The ignore changes specification is created.

What to do next

You can now reference this ignore changes specification when you compare objects.

Related information

Managing ignore changes (IBM Db2 Administration Tool for z/OS 13.1.0)

Modifying ignore changes specifications

You can update an *ignore changes specification*, which is a list of object changes that are to be ignored during the compare process.

Procedure

To modify ignore changes specifications:

1. On the **DB2 Object Comparison Tool Menu (GOCMENU)** panel, specify option 4, and press Enter.
2. On the **Specify Compare Ignores (GOC4)** panel, specify one of the following sets of information:

Edit preference	Fields to specify
To edit the specification as a list:	Under **Ignore Changes Specification**, specify an owner and name for the existing ignore changes specification and specify YES for **Edit Ignore Changes Specification**.
To edit the compare result with the ignore changes marked:	Under **Saved Compare Results**, specify an owner and name for the saved result, and specify YES for **Display using a saved compare result** .

```
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 57. **Specify Compare Ignores (GOC4)** panel*

3. If you specified YES for **Edit Ignore Changes Specification**, modify the list as needed on the **Ignored Changes List (ADBPCICL)** panel:

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

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

       Target    Target
Sel T  Qual      Name                Attribute       Additional Info

--- -- --------> ------------------> --------------- ----------------->

       DB23367   TS23367A            BUFFERPOOL
    TB VNDRG     TB23367             Data type       LAHMANID
    TB VNDRG     TB23367             Data type       DATE_YYYY
```

*Figure 58. **Ignored Changes List (ADBPCICL)** panel*

You can add object changes or delete an object change that is listed. When you add object changes, those changes are ignored. Ignored changes are not applied to the target objects.

To add an object change, add the object type and name in the blank line below the column headings and before the listed change objects. Use the wildcard character (*) in the **Target Qualifier** or **Target Name** column to indicate that all changes for matching objects are to be ignored. For example, if you specify new*, objects that meet the wildcard specification new* are still processed; however, all changes for these objects are ignored. If the qualifier or name does not include a wildcard character, the wildcard character (*) is appended to the qualifier or name. If the **Target Qualifier** or **Target Name** column is blank, an asterisk ('*') is substituted.

4. If you specified YES for **Display using a saved compare result** , use the line commands to modify the ignore changes as needed on the **Compare Report (ADBPCRR)** panel. See step "4" on page 122 in "Creating ignore changes specifications" on page 122.

Related tasks

“Creating ignore changes specifications” on page 122
An *ignore changes specification* designates the changes that you want ignored in subsequent compare processing.

Chapter 6. Applying changes to target objects

After you run a comparison, you can apply the changes in the compare report to the target object or objects.

About this task

Object Comparison Tool supports changes to implicit LOB and XML table spaces. When tables have implicit LOB or XML table spaces defined, Object Comparison Tool generates multiple image copies, which requires that a template be used for the SYSCOPY data set. You can define your own SYSCOPY template or Object Comparison Tool can use the default. The default template is:

```
DSN(&US..&SSID..&DB..&SN..&UQ)
```

The default template for clones is:

```
DSN(&US..&SSID..&DB..&SN..CLONE.&UQ)
```

For more information about templates and how to associate them with certain data sets, see Associating templates with data sets (IBM Db2 Administration Tool for z/OS 13.1.0).

Procedure

To apply changes to the target objects:

1. On the **DB2 Object Comparison Tool Menu (GOCMENU)** panel, select option **5 - Generate compare job**, and press Enter.
2. On the **Generate Compare Jobs (GOC5)** panel, set the **Generate Apply Job** field to Yes and set any other fields as needed, and press Enter.
3. On the **Specify Data Set Name for Apply Jobs (GOC5AJ)** panel, enter the name of the data set where you want the apply jobs generated.
4. Edit the generated compare job as needed, and submit the job.
5. Check the output to confirm that the job completed successfully.
6. Run the generated apply job to make the changes to your target objects or if you requested a work statement list (WSL), Run the WSL to apply the changes.

Running a work statement list to apply changes

Apply changes are placed in a work statement list (WSL) if you specified **Generate apply jobs** = YES and **As work statement list** = YES on the **Generate Compare Jobs (GOC5)** panel. You must use Db2 Admin Tool panels to run the work statement list.

About this task

When you run a WSL, you can generate single or multiple apply jobs for all operations. 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 WSL. 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 job be generated when you run the WSL. If you specify one job, table GRANT statements are generated by Object Comparison Tool after tables, related indexes, and foreign keys are created, and after all rows are reloaded.

Note: In the LOAD step, the TEMPLATE statement for the SYSREC data set is used when you select HPU for the unload process. Because Object Comparison Tool does not know if HPU will be used as the unload method before the WSL is run, the template might seem unneeded when the WSL is generated. However, the TEMPLATE SYSREC statement is used if you select HPU as the unload process when you build the job in Db2 Admin Tool.

Procedure

Use Db2 Admin Tool. See Running a WSL(IBM Db2 Administration Tool for z/OS 13.1.0).

Related reference

“Compare job options” on page 80
When you generate a compare batch job, you can specify a number of options to control the behavior of the comparison operation and job. These options are listed on the **Generate Compare Jobs (GOC5)** panel.

Related information

Work statement lists (IBM Db2 Administration Tool for z/OS 13.1.0)

Chapter 7. Converting version files to the latest Db2 version

To save time during compare processing, you can convert an older version file to the latest Db2 version that is supported by Db2 Object Comparison Tool by using the Version File Conversion utility. For example, if a version file was created for Db2 11 and you are running Db2 12, you can convert the version file for Db2 12.

About this task

A *version file* contains object information that is extracted by Db2 Object Comparison Tool during the compare process; this file is then used during the comparison.

Each version of Db2 Admin Tool and Object Comparison Tool supports a range of version file levels. If an unconverted version file that is within the supported range is passed to Object Comparison Tool, the tool can process the version file but requires extra conversion time. The original version file is unchanged by Object Comparison Tool

Alternatively, the Version File Conversion utility converts a version file within the supported range to the current level before it is processed. A converted version file can be used by Object Comparison Tool immediately, saving the time that the conversion would require. This utility converts the version file catalog records to the latest supported version of Db2 and permanently upgrades the file to the current supported level.

You can use the Version File Conversion utility to convert version files that are stored in sequential data sets, PDS data sets, and PDSE data sets and base versions that are stored using Db2 Admin Tool.

Procedure

To convert version files to the latest Db2 version, run the Version File Conversion utility by using a batch job. Sample JCL is provided in the SAMP library ADBVFCON member. Modify this sample as follows:

- Tailor the JCL to your installation before use.
- Specify input to the Version File Conversion utility by 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(SOURCEOO)', VFNEW='C386799.DT27760.C.SOURCENN';
VFOLD='Z9.ORDER.VERSION(MAY29)', 'Z9.ORDER.VERSION(JUNE17)';
```

- Ensure that any sequential data sets named are allocated and cataloged first. If VFOLD or VFNEW are PDS data sets, the member names must be included. The VFNEW PDS member name can be a new name; however, VFOLD and VFNEW cannot have the same name. The data set must not contain any sequence numbers in columns 72 - 80.
- Use one of the following two ways to convert a base version file: by VOWNER and VNAME or by VID.

  Change Management option 4.1 lists all base version files that are stored in Db2 Admin Tool. VOWNER, VNAME, and VID 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.
- To convert a base version file and store it in a sequential, PDS, or PDSE data set, specify the base version file by using VOWNER and VNAME or by using VID and specify the output data set by using VFNEW. 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.

Results

The Version File Conversion utility 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.

Chapter 8. Comparing multiple sources and targets

You can compare sources and targets from one or more saved dialogs. For example, you can compare multiple saved targets to one saved source. This functionality is called *MultiCompare*.

Before you begin

You must have saved dialogs from previous comparisons. See “Saving dialogs” on page 110.

Procedure

To compare multiple sources and targets:

1. On the **DB2 Object Comparison Tool Menu (GOCMENU)** panel, specify option **MC**, and press Enter.
2. On the **Saved Dialogs (GOCMC1)** panel, specify the prefix of the data set where the dialogs are saved and press Enter.

```
Compare ------------------------- Saved Dialogs ----------------------- 14:05
Option ===>

Enter the prefix for saved dialog data sets:
  Prefix ===> RAJESHR.ADMIN.SAVEDLGS
```

*Figure 59. **Saved Dialogs (GOCMC1)** panel*

3. On the **Manage Dialogs MultiCompare (GOCMC)** panel, select the dialogs that you want to include in the comparison by using one of the following commands:

   RUN
   Runs the compare process for all included dialogs [those with a Y in the **I (Indicator)** column].

   Before issuing RUN, use the IC (include) and XC (exclude) line commands to ensure that the dialogs you want are included.

   RUNALL
   Runs the compare process for all the listed dialogs except the excluded dialogs [those with a N in the **I (Indicator)** column].

   Before issuing RUNALL, use the IC (include) and XC (exclude) line commands to ensure that the dialogs that you do not want to use are excluded.

   Attention: RUN and RUNALL will fail if any of the selected dialogs have an incomplete status.

```
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 60. **Manage Dialogs MultiCompare (GOCMC)** panel*

For information about this panel, see “Manage Dialogs MultiCompare (GOCMC) panel” on page 133.

4. On the **MultiCompare / Select Dialog (GOC2MCMC)** panel, specify whether you want to compare one source to multiple targets.

```
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 61. **MultiCompare / Select Dialog (GOC2MCMC)** panel*

   - To compare one source to multiple targets, specify Y and the dialog name that contains the source that you want to use, and press Enter. The source in the specified dialog is used in each compare process instead of the source in the saved dialogs.
   - To run Compare with the source and target that are defined in each saved dialog, specify N, and press Enter. The source that is specified for each of the saved dialogs is used.
5. On the **Generate Compare Jobs (GOC5)** panel, specify the compare job options, and press Enter.

   Consider the following restrictions and behavior:

   - The **Generate online** option is automatically set to No. Generating a work statement list online is not supported with MultiCompare.
   - If the **Single compare job** option is Yes, the number of steps in the job exceeds 255, and the **Member name** value exceeds six characters, the member name is truncated to allow the addition of a numeric suffix.
   - If the **Single compare job** option is No and the **Member name** value exceeds five characters, the member name is truncated to allow the addition of a numeric suffix. This suffix indicates 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 Name`*nnn*, where *nnn* uniquely identifies the work statement list for each target.

   For descriptions of all of the options on the **Generate Compare Jobs (GOC5)** panel, see "Compare job options" on page 80.
6. Complete the options on any subsquent pop-up windows.
7. Edit the generated JCL job as needed and submit it to run the comparison.

Manage Dialogs MultiCompare (GOCMC) panel

The **Manage Dialogs MultiCompare (GOCMC)** panel displays a list of saved dialogs. You can select from this list dialogs that you want to use in a comparison.

```
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 62. **Manage Dialogs MultiCompare (GOCMC)** panel*

This panels lists the following information for each 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 that the dialog was created.

Id
The user ID of the person who created the dialog.

I (Indicator)
An indication (Y or N) of whether 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 line commands.

Status
The status of the dialog. A status of INCOMPLETE 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**I (Indicator)** field is set to Y.

From this panel, you can perform the following actions on a dialog by using the listed line command:

S - Select
Selects the dialog to be included in the comparison.

D - Delete
Deletes the dialog from the library.

I - Interpret
Displays the source and target details for the dialog, for example:

```
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 63. **Interpret Dialog (GOCMI)** panel*

This panel indicates whether the source and target definitions are from DDL or the Db2 catalog. If the source or target is from the Db2 catalog, this panel lists all objects in the definition and their type.

RL - Repeat/change location
Replicates a dialog and changes the location of the target. When you specify RL, the **Distributed DB2 Systems (ADB2DDF)** panel displays the remote Db2 subsystems that are available from the Db2 subsystem that you are currently on:

```
DB2 Admin ----------------- Distributed DB2 Systems ---------------------------
Command ===>                                                 Scroll ===> PAGE

Select by typing '+'
Select the location you wish to use:                    DB2 System: DD1A
                                                        DB2 SQL ID: ADM001
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 64. **Distributed DB2 Systems (ADB2DDF)** panel*

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. Press End. The **Repeat Dialog/Change Location (GOCMCRL)** panel displays the new dialog with the new location:

```
Compare--------------------- Repeat Dialog/Change Location --- Row 1 to 1 of 1
Command ===>                                                 Scroll ===> PAGE

Commands: NEXT

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 65. **Repeat Dialog/Change Location (GOCMCRL)** panel*

You can select and edit the dialog name and description. When you are finished editing, issue the NEXT command to return to the **Manage Dialogs MultiCompare (GOCMC)** panel.

R*nn* - Repeat *nn* times
Replicates a dialog multiple times. The **Repeat Dialog/Change Location (GOCMCRL)** panel displays 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 NEXT command to return to the **Manage Dialogs MultiCompare (GOCMC)** panel.

MS - Modify Source
Modifies the source details. The **Specify Compare Source (GOC1)** panel is displayed.

MT - Modify Target
Modify the target details. The **Specify Compare Target (GOC1)** panel is displayed.

IC - Include in Compare
Selects the dialog to include in the compare process. Upon selection, the **I (Indicator)** field is set to Y. The dialog status must be `Complete` for it to be included in the compare process; otherwise, an error occurs.

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 **I (Indicator)** field from Y to N.

Related tasks

“Comparing multiple sources and targets” on page 131
You can compare sources and targets from one or more saved dialogs. For example, you can compare multiple saved targets to one saved source. This functionality is called *MultiCompare*.

“Saving dialogs” on page 110

Chapter 9. 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 9 on page 50) 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

Related concepts

“Components of the comparison process ” on page 10
Db2 Object Comparison Tool compares objects by reading the Db2 catalog or DDL files. Object Comparison Tool produces comparison reports and then optionally generates either JCL jobs or work statement list (WSL) tasks with changes for the target objects.

“Batch compare program ” on page 145
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 a DDL file for the source or target definition” on page 49
Db2 Object Comparison Tool can use a file that contains data definition language (DDL) for the definitions of the source or target objects. Object Comparison Tool processes everything in the DDL file; objects are not selected based on type or name. If your DDL defines a single table, only that table is used.

Related reference

“Supported SQL statements ” on page 137
The DDL file extraction program supports a subset of the SQL statements that are supported by DB2 for z/OS.

“Batch DDL file extraction program report format ” on page 140
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.

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 Tool Version 11.1 and is the ` (grave accent) for Db2 Admin Tool 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 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 66. CREATE VERSION report from DDL file

Chapter 10. 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.

Related tasks

"2. Specifying target objects" on page 60
After you specify the comparison source, the next step is to specify the target. The *target* is the object or objects that you want to compare to the source.

Related reference

"Batch DB2 catalog extraction program report" on page 141
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.

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 67. 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.

```
>-------------------------------------------------------------------------------------------------------
ADB2GEN - Create DDL from catalog info                                              2006-11-29 13:50
-------------------------------------------------------------------------------------------------------


      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                                              2006-11-29 13:50
-------------------------------------------------------------------------------------------------------

      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
-------------------------------------------------------------------------------------------------------
ADB2GEN - Create DDL from catalog info                                              2006-11-29 13:50
-------------------------------------------------------------------------------------------------------
```

Figure 68. DSNHDECP values

Chapter 11. 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 "5. Generating a compare batch job" on page 78 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 "Excluding objects from the compare process" on page 117 for more information about creating and using Exclude Specification.

Refer to Chapter 13, "Batch compare report format ," on page 163 for batch compare report examples.

Related concepts

"Components of the comparison process " on page 10
Db2 Object Comparison Tool compares objects by reading the Db2 catalog or DDL files. Object Comparison Tool produces comparison reports and then optionally generates either JCL jobs or work statement list (WSL) tasks with changes for the target objects.

"Batch DDL file extraction program " on page 137
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

"5. Generating a compare batch job" on page 78
A *compare batch job* is a JCL job that performs the requested comparison.

"Specifying a version scope for the source or target definition" on page 59

Related reference

"Batch compare report format " on page 163
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

Db2 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 (GOC5)** panel.

- 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 object

  The 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

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 (GOC5)** panel.

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 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 "3. Specifying compare masks" on page 63. 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 "3. Specifying compare masks" on page 63 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

Restrictions:

- 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.
- Renaming an implicit index is not supported.
- Renaming 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 (GOC5)** panel.

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.

For more information, see the *DB2 Universal Database for z/OS SQL Reference*.

Table drop/re-create without data conversion

Under certain conditions Db2 Object Comparison Tool 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 7. 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	–	–
DECFLOAT(34)	Y	Y	Y	Y	–	–	–	–	–	–	–	–	Y	Y	–	–
BINARY	–	–	–	–	Y	–	–	–	–	–	–	–	–	–	Y	–
VARBINARY	–	–	–	–	–	Y	–	–	–	–	–	–	–	–	–	Y

Table 7. 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. *(continued)*

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

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 (GOC5)** panel. The following table provides more information.

Table 8. 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.
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.

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 12. Running Compare by using a Change Management batch job

You can use the Db2 Admin Tool Change Management (CM) batch interface to run Db2 Object Comparison Tool in batch. By using this interface, you can define or propagate a change that can be managed by Db2 Admin Tool Change Management.

Procedure

To run Compare by using CM batch:

- Create and run a CM batch job with the following parameters and specifications:

  Required parameters:

CM batch parameter	Purpose	Link to parameter information
`ACTION_COMPARE='Y'`	Indicates that you want to run Compare.	ACTION_COMPARE (IBM Db2 Administration Tool for z/OS 13.1.0)
SOURCE_TYPE	Specifies the compare source. You must specify any other parameters that are required based on the type value that you specify.	SOURCE_TYPE (IBM Db2 Administration Tool for z/OS 13.1.0)
TARGET_TYPE	Specifies the compare target. You must specify any other parameters that are required based on the type value you specify.	TARGET_TYPE (IBM Db2 Administration Tool for z/OS 13.1.0)

  Optional parameters:

CM batch parameter[1]	Purpose	Link to parameter information
`ACTION_IMPORT_CHANGE='N'`	Prevents the generated delta change file from being imported as a new registered change and analyzed on the local system. (This behavior is the default.)	ACTION_IMPORT_CHANGE (IBM Db2 Administration Tool for z/OS 13.1.0)
Any other relevant CM batch parameter	See the description for each parameter.	CM batch parameter definitions (IBM Db2 Administration Tool for z/OS 13.1.0)

Other optional specifications:

Item to specify	Method	Link to parameter information
Compare masks	Specify any compare masks by taking one of the following actions: – 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.	COMPARE_MASK_DSN (IBM Db2 Administration Tool for z/OS 13.1.0) COMPARE_MASK_OWNER (IBM Db2 Administration Tool for z/OS 13.1.0) COMPARE_MASK_NAME (IBM Db2 Administration Tool for z/OS 13.1.0)
Ignore fields	Specify any ignore fields by taking one of the following actions: – Use the **COMPARE_IGNORE_FIELDS_DSN** parameter to specify the name of an existing data set that contains the compare ignore fields. – 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. – Preallocate compare ignore fields file with DD name of IGNORES.	COMPARE_IGNORE_FIELDS_DSN (IBM Db2 Administration Tool for z/OS 13.1.0) COMPARE_IGNORE_FIELDS_OWNER (IBM Db2 Administration Tool for z/OS 13.1.0) COMPARE_IGNORE_FIELDS_NAME (IBM Db2 Administration Tool for z/OS 13.1.0)
Ignore changes	Specify any ignore changes by using the **COMPARE_IGNORE_CHANGES_OWNER** and **COMPARE_IGNORE_CHANGES_NAME** parameters. These parameters identify an existing ignore changes specification that is stored in the Change Management database.	COMPARE_IGNORE_CHANGES_OWNER (IBM Db2 Administration Tool for z/OS 13.1.0) COMPARE_IGNORE_CHANGES_NAME (IBM Db2 Administration Tool for z/OS 13.1.0)

Item to specify	Method	Link to parameter information
Excludes	Specify an existing exclude specification that is stored in the Change Management database as follows: – For the compare source, use the parameters **SOURCE_EXCLUDE_OWNER** and **SOURCE_EXCLUDE_NAME**. – For the compare target, use the parameters **TARGET_EXCLUDE_OWNER** and **TARGET_exclude_NAME**.	SOURCE_EXCLUDE_OWNER (IBM Db2 Administration Tool for z/OS 13.1.0) SOURCE_EXCLUDE_NAME (IBM Db2 Administration Tool for z/OS 13.1.0) TARGET_EXCLUDE_OWNER (IBM Db2 Administration Tool for z/OS 13.1.0)TARGET_EXCLUDE_NAME (IBM Db2 Administration Tool for z/OS 13.1.0)

Results

The compare report and a delta change file that describes the differences is generated.

Examples of using SELECT statements to identify source and target objects

The following sample lines from CM batch jobs show examples of CM batch parameter specifications:

Specifying the SELECT statement in the SQL parameter

The following lines request that the specified SQL SELECT statement (in TGTIN DD) be used to select the target objects. Those objects are to be compared with the objects identified by the DDL in the specified source data set, SOURCE_DATASET_NAME (in SRCIN DD).

```
//GOCCM.PARMS DD *
CHANGE_NAME                 = 'CB315061'
ACTION_ANALYZE_CHANGE       = 'Y'
ACTION_RUN_CHANGE           = 'Y'
ACTION_COMPARE              = 'Y'
ACTION_IMPORT_CHANGE        = 'Y'
EXISTING_DATA_SET_ACTION    = 'REPLACE'
TARGET_TYPE              = 'USER'
SOURCE_TYPE              = 'DDL'
SUPPRESS_DROP_OF_OBJECTS    = 'YES'
PREFIX_FOR_DATA_SETS        = '&TSOID'
REPORT_SUMMARY              = 'Y'
REPORT_ONLY_CHANGED_OBJECTS = 'N'
REPORT_OBJECT_TOTALS        = 'Y'
/*
//ADBMSGS  DD SYSOUT=*
//SRCIN DD DSN=SOURCE_DATASET_NAME,DISP=SHR
//TGTIN DD *
TYPE='SQL',SQL="
SELECT 'TS' AS TYPE,
DBNAME AS QUAL,
NAME FROM SYSIBM.SYSTABLESPACE
WHERE NAME LIKE 'TR31506%'";
//*
```

Specifying the name of that data set that contains the SELECT statement

The following lines request that both the target and source objects be selected by the given SQL SELECT statements. For the target, the SELECT statement is listed in the SQL parameter (in TGTIN DD). For the source, the SELECT statement is listed in another data set, which is identified by the DSN parameter in the SRCIN DD statement.

```
//GOCCM.PARMS DD *
ACTION_COMPARE             = 'Y'
ACTION_IMPORT_CHANGE       = 'N'
ADBTEP2_RESTART            = 'N'
ACTION_GENERATE_WSL        = 'Y'
ACTION_RUN_WSL             = 'Y'
REPORT_SUMMARY             = 'Y'
REPORT_OBJECT_TOTALS       = 'Y'
RUN_REORG_REBUILD          = 'A'
TARGET_TYPE              = 'USER'
SOURCE_TYPE              = 'USER'
SUPPRESS_DROP_OF_OBJECTS   = 'YES'
EXISTING_DATA_SET_ACTION      = 'REPLACE'
PREFIX_FOR_DATA_SETS          = '&TSOID'
PDS_FOR_WSL                   = '&SSID.WSL'
WORKLIST_NAME                 = 'CB315065';
/*
//GOCCM.REPORT   DD SYSOUT=*
//GOCCM.ADBDIAG  DD SYSOUT=*
//GOCCM.ADBMSGS  DD SYSOUT=*
//GOCCM.SYSPRINT DD SYSOUT=*
//GOCCM.MASKS    DD *
DBNAME:DB315063,DB315061
TSNAME:TR315063,TR315061
TBNAME:TB315063,TB315061
IXNAME:IX315063,IX315061
IXNAME:IX315064,IX315062
//GOCCM.IGNORES  DD *
  SYSDATABASE: BPOOL
//TGTIN DD *
TYPE='SQL', SQL="SELECT 'DB' AS TYPE, '' AS QUAL,
NAME FROM SYSIBM.SYSDATABASE WHERE NAME='DB315061'";
//GOCCM.SRCIN DD DSN=SOURCE_DATASET_NAME,DISP=SHR
/*
```

SOURCE_DATASET_NAME contains:

```
TYPE='SQL',
SQL="SELECT 'DB' AS TYPE, '' AS QUAL,
NAME FROM SYSIBM.SYSDATABASE
WHERE NAME='DB315063'"
;
```

Specifying a DD name for the data set that contains the SELECT statement

The following lines request that both the target and source objects be selected by the given SQL SELECT statements. For the source, the SELECT statement is listed in the SQL parameter (in SRCIN DD). For the target, the SELECT statement is listed in another data set, and the SQL parameter (in TGTIN DD) lists the DD name. That referenced DD statement identifies the data set that contains the SQL statement.

```
//GOCCM.PARMS DD *
ACTION_COMPARE             = 'Y'
ACTION_IMPORT_CHANGE       = 'N'
ADBTEP2_RESTART            = 'N'
ACTION_GENERATE_WSL        = 'Y'
ACTION_RUN_WSL             = 'Y'
REPORT_SUMMARY             = 'Y'
REPORT_OBJECT_TOTALS       = 'Y'
RUN_REORG_REBUILD          = 'A'
TARGET_TYPE              = 'USER'
SOURCE_TYPE              = 'USER'
SUPPRESS_DROP_OF_OBJECTS   = 'YES'
EXISTING_DATA_SET_ACTION      = 'REPLACE'
PREFIX_FOR_DATA_SETS          = '&TSOID'
PDS_FOR_WSL                   = '&SSID.WSL'
WORKLIST_NAME                 = 'CB315064';
/*
//GOCCM.REPORT   DD SYSOUT=*
//GOCCM.ADBDIAG  DD SYSOUT=*
//GOCCM.ADBMSGS  DD SYSOUT=*
//GOCCM.SYSPRINT DD SYSOUT=*
//GOCCM.MASKS    DD *
DBNAME:DB315063,DB315061
TSNAME:TR315063,TR315061
TBNAME:TB315063,TB315061
IXNAME:IX315063,IX315061
IXNAME:IX315064,IX315062
```

```
//SRCIN DD *
TYPE='SQL', SQL="
SELECT 'TS' AS TYPE,
DBNAME AS QUAL,
NAME FROM SYSIBM.SYSTABLESPACE
WHERE NAME LIKE 'TR315063'";
//TGTIN DD *
   TYPE='SQL', SQL='=SQLTRG';
//SQLTRG DD DSN=TARGET_DATASET_NAME,DISP=SHR
/*
```

Related concepts

"Translation masks" on page 65
In Object Comparison Tool, you can use translation masks to account for differences in naming conventions between source and target objects when doing a comparison. You can also use masks to overwrite values for object attributes.

Related information

Managing Changes by using the CM batch interface (IBM Db2 Administration Tool for z/OS 13.1.0)

CM batch parameter definitions (IBM Db2 Administration Tool for z/OS 13.1.0)

Comparing table pairs

You can compare regular tables with an archive-enabled and archive table pair. The following table shows the compare results for various sources and targets. B1 and B2 are regular tables that have corresponding archive and archive-enabled tables.

Table 9. Archive table compare results

Source	Target	Expected result
None	Archive-enabled table (B1) and archive table (B2)	Archive-enabled table (B1) is dropped. Archive table (B2) will be dropped as a result of dropping the archive enabled table.
Table B1	Archive-enabled table (B1) and archive table (B2)	Table B1 is compared. ALTER TABLE B1 DISABLE ARCHIVE is generated.
Table B2	Archive-enabled table (B1) and archive table (B2)	ALTER TABLE B1 DISABLE ARCHIVE is generated. Table B1 is dropped. Table B2 is compared.
Table B1 Table B2	Archive-enabled table (B1) and archive table (B2)	Table B1 and B2 are compared. ALTER TABLE B1 DISABLE ARCHIVE is generated.
Archive-enabled table (B1) and archive table (B2)	None	Table B1 and B2 are added. ALTER TABLE B1 ENABLE ARCHIVE USE B2 is generated.

Table 9. Archive table compare results (continued)		
Source	**Target**	**Expected result**
Archive-enabled table (B1) and archive table (B2)	Table B1	Table B1 is compared. Archive table (B2) is added. ALTER TABLE B1 ENABLE ARCHIVE USE B2 is generated.
Archive-enabled table (B1) and archive table (B2)	Table B2	Table B1 is added. Table B2 is compared. ALTER TABLE B1 ENABLE ARCHIVE USE B2 is generated.
Archive-enabled table (B1) and archive table (B2)	Table B1 Table B2 (no archive relationship between these two tables)	Table B1 is compared. Table B2 is compared. ALTER TABLE B1 ENABLE ARCHIVE USE B2 is generated.

You can also compare temporal and history table pairs. The following table shows the compare results for various sources and targets. B1 and B2 are regular tables that have corresponding temporal and history tables.

Table 10. Temporal-history table compare results		
Source	**Target**	**Expected result**
None	Temporal table (B1) and history table (B2)	Temporal table (B1) is dropped. History table (B2) will be dropped by dropping the history-enabled table.
Table B1	Temporal table (B1) and history table (B2)	Table B1 is compared. ALTER TABLE TEMPORAL TABLE DROP VERSIONING is generated.
Table B2	Temporal table (B1) and history table (B2)	ALTER TABLE B1 DROP VERSIONING is generated. Table B1 is dropped. Table B2 is compared.
Temporal table and history table (no temporal-history relationship between these two tables)	Temporal table (B1) and history table (B2)	Table B1 and B2 are compared. ALTER TABLE B1 DROP VERSIONING is generated.
Temporal table (B1) and history table (B2)	None	Table B1 and B2 are added. ALTER TABLE B1 ADD VERSIONING USE B2 is generated.

Table 10. Temporal-history table compare results (continued)		
Source	**Target**	**Expected result**
Temporal table (B1) and history table (B2)	Table B1	Table B1 is compared. History table (B2) is added. ALTER TABLE B1 ADD VERSIONING USE B2 is generated.
Temporal table (B1) and history table (B2)	Table B2	Table B1 is added. Table B2 is compared. ALTER TABLE B1 ADD VERSIONING USE B2 is generated.
Temporal table (B1) and history table (B2)	Table B2 Table B2 (no temporal-history relationship between these two tables)	Table B1 is compared Table B2 is compared. ALTER TABLE B1 ADD VERSIONING USE B2 is generated.

Chapter 13. 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 to be 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)                      |
    |      Object specific . . . . : 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 69. 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.

  Important: 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 sections:

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, 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

  The objects that are not found in the target are reported as added objects. In this case, objects are added. For example, the report shows that a new view named VNDR230.VDEPTS is to be added.
- Dropped objects

  The objects that are not found in the source are reported as dropped objects. 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 compared 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 to apply changes” on page 127.

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, 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

The count report shows how many objects were processed per object type. It 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 and how many were dropped and recreated.

Specify Yes or No to indicate whether you want the report to include a comparison counts 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.

Tip: If the report includes unexpected changes to bind options for trigger packages, you might need to rebind some packages. For detailed information, see “Troubleshooting: The Compare report shows changes to bind options for trigger packages” on page 204.

Related concepts

“Batch compare program ” on page 145
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

“Compare job options” on page 80
When you generate a compare batch job, you can specify a number of options to control the behavior of the comparison operation and job. These options are listed on the **Generate Compare Jobs (GOC5)** panel.

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.

```
scale="80">--------------------------------------------------------------------------------------------------------
-------
 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
 ========================================================


 SYSAUXRELS      : 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),
                   COPYLRSN(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), 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(U), CREATEDTS(S), ALTEREDTS(S),
                   RBA1(S), RBA2(S), PCTROWCOMP(S), STATSTIME(S), CARDF(S),
                   CHECKRID5B(S), NPAGESF(S), SPACEF(S), AVGROWLEN(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)
 SYSVOLUMES      : 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
```

```
Recreated
 ------------------   ----------  ----------  ----------  ----------  ----------  ---------- ----------
----------
 Tablespaces                   1           1           1           0           0           0
0          0
 Tables                        1           1           1           0           0           0
0          0
 Indexes                       3           3           3           0           0           0
0          0
 Views                         8           7           7           1           0           0
0          0
 Relations                     2           2           2           0           0           0
0          0
```

If the comparison summary report includes any object names that are longer than 18 characters, those names are displayed across multiple rows, in the same column and with the same indentation, as shown in the following example:

```
COMPARISON SUMMARY REPORT
 ========================

 Obtyp Source Object                    Target Object                    Result               Object type
 ----- -------------------------------  -------------------------------  -------------------- ------------
 D     TESTECUS                         TESTECUS                         No change            Database
  S     TESTECUS.SH4909AB                TESTECUS.SH4909AB                No change            Tablespace
   T     VIV1.EC_DOC_ATTRIBUTEDDDDEC_DO   VIV1.EC_DOC_ATTRIBUTEDDDDEC_DO Altered              Table
         C_ATTRIBUTEDDDDFOGHTRESDFREEC_   C_ATTRIBUTEDDDDFOGHTRESDFREEC_
         DOC_ATTRIBUTEDDDDEC_DOC_ATTRIB   DOC_ATTRIBUTEDDDDEC_DOC_ATTRIB
         UTEDDDDEC_DOC_ATTRIBUTED112ABC   UTEDDDDEC_DOC_ATTRIBUTED112ABC
         DEABCDEDRASDF                    DEABCDEDRASDF
    X     VIV1.SH4909ABA1SH4909ABA1SH49    VIV1.SH4909ABA1SH4909ABA1SH49 No change            Index
    X     VIV1.SH4909ABA2SH4909ABA2SH49    VIV1.SH4909ABA2SH4909ABA2SH49 No change            Index
          0                                0
    X     VIV1.SH4909ABA3SH4909ABA3SH49                                  Added                Index
          09ABA3SH4909ABA3ASDFGSH4909AB
          A3SH4909ABA3SH4909ABA3SH4909A
          BA3SH4909ABA3SH4909ABA3SH4909
          ABA3FGSH4909
    X     VIV1.SH4909ABA4SH4909ABA4SH4S                                  Added                Index
          H
    X     VIV1.SH4909ABPK                    VIV1.SH4909ABPK              No change            Index
  S     TESTECUS.SH4914                  TESTECUS.SH4914                  No change            Tablespace
   T     VIV1.EC_ATTRIBUTE_TYP                                           Added                Table
    X     VIV1.SH4914PK                      VIV1.SH4914PK                Altered              Index
   T                                       VIV1.EC_ATTRIBUTE_TYP_CD       Dropped              Table
   T                                       VIV1.EC_ATTRIBUTE_TYP_CD_TABLE Dropped              Table
                                           WITH_LONG_NAME_TESTINGSUITE
 ----- -------------------------------  -------------------------------  -------------------- ------------
```

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.

```
scale="80">-------------------------------------------------------------------------------------------------------
-------
 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
 =========================
```

```
 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               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
Recreated
 ------------------   ----------  ----------  ----------  ----------  ----------  ----------  ----------
----------
 Schemas                      0           0           0           0           0           0
0          0
 User Def Types               0           0           0           0           0           0
0          0
 Sequences                    0           0           0           0           0           0
0          0
 Databases                    0           0           0           0           0           0
0          0
 Tablespaces                  1           1           1           0           0           0
0          0
 Tables                       1           1           1           0           0           0
0          0
 Indexes                      3           3           3           0           0           0
0          0
 Aliases                      0           0           0           0           0           0
0          0
 Storage groups               0           0           0           0           0           0
0          0
 Synonyms                     0           0           0           0           0           0
0          0
 Functions                    0           0           0           0           0           0
0          0
 Stored procedures            0           0           0           0           0           0
0          0
 Triggers                     0           0           0           0           0           0
0          0
 Views                        3           7           2           1           5           0
0          0
 Relations                    2           2           2           0           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.

```
scale="80">-------------------------------------------------------------------------------------------------
-------
 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


SYSAUXRELS      : 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)
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),
                  COPYLRSN(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), AVGROWLEN(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   Recreated
 ------------------   ----------  ----------  ----------  ----------  ----------  ----------
----------  ----------
```

```
 Tablespaces                    1           1           1           0           0           0
0           0
 Tables                         1           1           1           0           0           0
0           0
 Indexes                        3           3           3           0           0           0
0           0
 Views                          8           7           7           1           0           0
0           0
 Relations                      2           2           2           0           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.

```
 scale="80">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
Recreated
------------------   ----------  ----------  ----------  ----------  ----------  ----------  ----------
----------
Schemas                       0           0           0           0           0           0
0           0
User Def Types                0           0           0           0           0           0
0           0
Sequences                     0           0           0           0           0           0
0           0
Databases                     1           1           1           0           0           0
0           0
Tablespaces                   7           4           4           1           0           1
2           0
Tables                        7           4           4           1           0           0
2           0
Indexes                       7           4           4           1           0           0
2           0
Aliases                       0           0           0           0           0           0
0           0
Storage groups                0           0           0           0           0           0
0           0
Synonyms                      0           0           0           0           0           0
0           0
Functions                     0           0           0           0           0           0
0           0
Stored procedures             0           0           0           0           0           0
0           0
Triggers                      0           0           0           0           0           0
0           0
Views                         0           0           0           0           0           0
0           0
Column masks                  0           0           0           0           0           0
0           0
Row permissions               0           0           0           0           0           0
0           0
Relations                     0           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 report 5

The sample batch compare report in the following figure shows a subset of the compare report for which the comparison was run for some specific objects, rather than all object types. You can generate reports in this format by entering the value Y for the Object specific option on the Specify Compare Reporting Options panel (GOC5RO).

```
>FIELDS IGNORED WHEN COMPARING SPECIFIC OBJECTS
==============================================

Specified Object Name                    Compared Object Name
---------------------------------------- ----------------------------------------
DB47985.TS47985A                         DB47985.TS47985A
(f)SYSTABLESPACE:PARTITIONS,BPOOL
DB47985.TS47985B                         DB47985.TS47985B
(f)SYSINDEXPART:PARTITION,LIMITKEY
(f)SYSTABLEPART:PARTITION,LIMITKEY,LIMITKEY_INTERNAL,LOGICAL_PART
(f)SYSTABLES:PARTKEYCOLNUM
(f)SYSTABLESPACE:PARTITIONS
(f)SYSCOLUMNS:PARTKEY_COLSEQ,PARTKEY_ORDERING
(f)SYSAUXRELS:PARTITION
DB47985.TS479*5B                         DB47985.TS47985B
(f)SYSTABLESPACE:PARTITIONS
DB47985.TS47985C                         DB47985.TS47985C
(f)SYSTABLESPACE:BPOOL
DB47985.TS47985D                         No Match
(f)SYSTABLESPACE:BPOOL
DB47985                                  DB47985
(f)SYSDATABASE:INDEXBP
DB47985.TS479*5B                         DB47985.TS47985B
(g)PBG_NUMPARTS
DB47985.TS47985*                         DB47985.TS47985A
                                         DB47985.TS47985B
                                         DB47985.TS47985C
(f)SYSTABLESPACE:TYPE,ENCODING_SCHEME,SBCS_CCSID,DBCS_CCSID,MAXROWS,
   LOCKPART,LOG,CURRENT_VERSION,CREATORTYPE,INSTANCE,CLONE
DB4798501234567890>.TS47985E9012345678> No Match
(f)SYSTABLESPACE:PARTITIONS
-
(g) Indicates the name of the GENERIC group that was used.
(f) Indicates table name:field list.
```

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 14. 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 70. 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.ALTENATE.ALTPDS(GOCSVARS)
//         DD DISP=SHR,DSN=HLQ.ALTENATE.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

If you specify an alternate data set for input to the program that generates the apply job, you can specify alternate values for the input variables.

The following 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 11. Alternate shared variable input data

Variable	Definition	Valid Input
AAPFLIBR	DB2 Admin APF authorized library.	A data set name. For example: DMTOOL.SADBLINK
ADB081CM	DB2 8 CM	Y or N. Specify Y if Db2 is at this release level or higher.
ADB081NF	DB2 8 NFM	Y or N. Specify Y if Db2 is at this release level or higher.
ADB091CM	DB2 9 CM	Y or N. Specify Y Db2 is at this release level or higher.
ADB091NF	DB2 9 NFM	Y or N. Specify Y if Db2 is at this release level or higher.
ADB101CM	Db2 10 CM	Y or N. Specify Y if Db2 is at this release level or higher.
ADB101NF	Db2 10 NFM	Y or N. Specify Y if Db2 is at this release level or higher.
ADB111CM	Db2 11 CM	Y or N. Specify Y if Db2 is at this release level or higher.
ADB111NF	Db2 11 NFM	Y or N. Specify Y if Db2 is at this release level or higher.
ADB121NF	Db2 12	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: 'DMTOOL.SGOCEXEC' 'DMTOOL.SADBEXEC'
ADBJ1	Job card line 1 (ADB2UPA).	A job card of up to 72 characters. Any valid job card syntax line.

Table 11. Alternate shared variable input data (continued)		
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.
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: 'DMTOOL.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=  ;`
		`CCCDD70154444444`
		`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=  ;`
		`CCCDDC70154444444`
		`142533ED5E0000000`

ADBPRIM	Primary space allocation (ADB2UPA)	A numeric value specified in &ADBSPAC 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 &ADBSPAC units.
ADBSPAC	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.

Table 11. Alternate shared variable input data (continued)		
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.
ADBTEST	Use test plan	YES or any other value.
ADBTLTB	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'.
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.

Table 11. Alternate shared variable input data (continued)		
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.
ASYRECDC	Use activate HPU Parallel Unload/ Load in the batch apply job (ADB2UCUS)	A template name. Associated with ALULD keyword on ADB2UCUS panel.
ASYSLLIA	ISPF linklist library 2	A data set name.
ASYSLLIB	ISPF linklist library 1	A data set name.
ASYSMLIB	ISPF message library	A data set name.
ASYSTLIB	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*).
CTNDISC1	DISCARDDN (ADB25TU4)	A template name. Used as the parameter to the DISCARDDDN keyword. For example: DISCARDDDN(*mytemp3*).
CTNERR	ERRDDN (ADB25TU4)	A template name. Used as the parameter to the ERRDDN keyword. For example: ERRDDN(*mytemp4*).
CTNFCOPY	FCCOPYDDN (ADB25TU4)	A template name. Used as the parameter to the FCCOPYDDN keyword. For example: FCCOPYDDN(*mytemp5*).
CTNFILTR	FILTERDDN (ADB25TU4)	A template name. Used as the parameter to the FILTERDDN keyword. For example: FILTERDDN(*mytemp6*).
CTNMAPDD	MAPDDN (ADB25TU4)	A template name. Used as the parameter to the MAPDDN keyword. For example: MAPDDN (*mytemp7*).
CTNPUNCH	PUNCHDDN (ADB25TU4)	A template name. Used as the parameter to the PUNCHDDN keyword. For example: PUNCHDDN(*mytemp8*).
CTNRECV1	RECOVERYDDN 1 (ADB25TU4)	A template name. Used as the first parameter to the RECOVERYDDN keyword. For example: RECOVERYDDN(*mytemp9*).
CTNRECV2	RECOVERYDDN 2 (ADB25TU4)	A template name. Used as the second parameter to the RECOVERYDDN keyword. For example: RECOVERYDDN(*mytemp9*, *mytempA*).
CTNUNLDD	UNLDDN (ADB25TU4)	A template name. Used as the parameter to the UNLDDN keyword. For example: UNLDDN(*mytempB*).
CTNWORK1	WORKDDN 1 (ADB25TU4)	A template name. Used as the first parameter to the WORKDDN keyword. For example: WORKDDN(*mytempc*).
CTNWORK2	WORKDDN 2 (ADB25TU4)	A template name. Used as the second parameter to the WORKDDN keyword. For example: WORKDDN(*mytempC*, *mytempD*).

Table 11. Alternate shared variable input data (continued)		
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 DISCARDDN (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.
DROPDSN	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 (GOC5AJ)	A data set name.
GOCAJOBN	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 ro specify None.
GOCGARUN	Run RUNSTATS (GOC5)	R – Reload A – Alter B – Both M – Minimum N – None.
GOCGAWL	As work statement list (GOC5)	Y or N.

Table 11. Alternate shared variable input data (continued)		
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(*mytempF*).
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 table space	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(*mytempE*).
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.
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.

Table 11. Alternate shared variable input data (continued)		
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 *devtype*.
USK05	SORTNUM	1-255. This value is used as a paramater 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.
USKS1	FOR EXCEPTION IN table schema	A schema name. This value is used in conjunction with &uskn1.

Table 11. Alternate shared variable input data (continued)		
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 paramter 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 *devt.*

Table 11. Alternate shared variable input data (continued)		
USO18	SORTNUM	1 - 255. This number is used as a parameter to SORTNUM. For example: SORTNUM 3.
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.	
USO40	LOGRANGES	• 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.
USO41	DRAIN_ALLPARTS	• 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.

Table 11. Alternate shared variable input data (continued)		
USO42	SWITCHTIME	• 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.
USO43	NEWMAXRO	• 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	• Y - YES • N - NO
USO45	LISTPARTS	• 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	• 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	• 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	• 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)	• Y - YES, gather information for all indexes on all tables in the table space • N - NO, do not gather such information
USO61	INDEX HISTOGRAM	• Y - YES, gather histogram statistics for all indexes on all tables in the table space • N - NO, do not gather such statistics

Table 11. Alternate shared variable input data (continued)		
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.
USORBALR	RBALRSN	Specifies the RBA and LRSN format in which the target object is to be left after a REORG. • 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, of 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 *devt.*
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.

Table 11. Alternate shared variable input data (continued)		
USR32	TABLESAMPLE	AUTO or numeric literal between '0.01' 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.*
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 *my.dsn*.
USU02	FROMVOLUME*	CATALOG or volid. This value is used as a parameter to FROMVOLUME. For example: FROMVOLUME *vol001*.
USU03	FROMCOPYDDN*	DD name. This value is used as a parameter to FROMCOPYDDN. For example: FROMCOPYDDN *dd001*.
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(*n*).
USU06	MIXED CCSID	A numeric value. This value is used as the second parameter to CCSID. For example: CCSID(1, *n*).
USU07	DBCS CCSID	A numeric value. This value is used as the third parameter to CCSID. For example: CCSID(1, 2, *n*).
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.

Table 11. Alternate shared variable input data (continued)		
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 ss used as a parameter to DECFLOAT_ROUNDMODE. For example: DECFLOAT_ROUNDMODE ROUND_UP.
USUUFI	FORMAT INTERNAL	Y or N. Use Y to specify FORMAT INTERNAL or N to specify no keyword.
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=' *PSV01*'.
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.

Table 11. Alternate shared variable input data (continued)		
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*.
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 &UTCRND 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.
UTCRND	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(*mytemp1*).
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(*mytemp1*, *mytemp2*).
UTNDISC1	Template discard data set name	A template name. This value is used as the parameter to the DISCARDDDN keyword. For example: DISCARDDDN(*mytemp3*).
UTNERR	Template error data set name (ADB25TU)	A template name. This value is used as the parameter to the ERRDDN keyword. For example: ERRDDN(*mytemp4*).
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(*mytemp5*).
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(*mytemp6*).
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 *(mytemp7*).

Table 11. Alternate shared variable input data (continued)		
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(*mytemp8*).
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(*mytemp9*).
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(*mytemp9*, *mytempA*).
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(*mytempB*).
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(*mytempc*).
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(*mytempC*, *mytempD*).
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.

Table 11. Alternate shared variable input data (continued)		
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.
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. • 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	• Y - YES, gather histogram statistics from the specified column group • N - NO, do not gather such statistics

Table 11. Alternate shared variable input data (continued)		
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	• 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	• 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.
UXO43	NEWMAXRO	• 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	Specifies the RBA and LRSN format in which the target object is to be left after a REORG. • N - None No conversion • B - Basic Convert to a basic format • E - Extended Convert to extended format

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. USERTEMPL 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_DISCARDDN_NAME**
- **UTIL_TEMPLATE_DISCARDDN_USE**
- **UTIL_CLONE_TEMPLATE_DISCARDDN_NAME**
- **UTIL_CLONE_TEMPLATE_DISCARDDN_USE**
- **UTIL_TEMPLATE_DISCARDDNC_NAME**
- **UTIL_TEMPLATE_DISCARDDNC_USE**
- **UTIL_CLONE_TEMPLATE_DISCARDDNC_NAME**
- **UTIL_CLONE_TEMPLATE_DISCARDDNC_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

Now you can 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 information

Using DB2 templates: Change Management (CM) batch interface (IBM Db2 Administration Tool for z/OS 13.1.0)

CM batch parameter definitions (IBM Db2 Administration Tool for z/OS 13.1.0)

Chapter 15. Recommendations when comparing a large number of objects

Enterprise Resource Planning (ERP) systems typically have a large number of objects. When you use Db2 Object Comparison Tool to compare a large number of objects, use the following recommendations:

- If online compare fails with an ONCODE=451 (out of storage), try running the comparison as a batch job. Using the same version files, a batch object compare job running under an initiator usually runs to completion. The online compare failure is due to TSO and ISPF control blocks, tasks and code that are not present in a batch address space, and limits on the region size for the TSO address space in which online compare is executing.
- Specify a large region size on the job card to ensure that the batch job can get sufficient virtual storage. If possible, specify 0M.
- 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 in advance.

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 16. Troubleshooting and messages

Use this information to diagnose and correct problems that you might experience when you customize Db2 Object Comparison Tool.

Messages with a prefix of ADB are from Db2 Admin Tool. For information about those messages, see Db2 Admin Tool messages (IBM Db2 Administration Tool for z/OS 13.1.0)

Messages with a prefix of CCQ are from Tools Customizer (TCz). For information about these messages, see Tools Customizer messages (IBM Tools Customizer for z/OS 1.1). For additional troubleshooting help with TCz, see Tools Customizer troubleshooting (IBM Tools Customizer for z/OS 1.1).

Related information

Gathering diagnostic information (IBM Db2 Administration Tool for z/OS 13.1.0)

Db2 Object Comparison Tool condition codes

Object Comparison Tool programs return condition codes to indicate whether the program completed successfully. If you receive a condition code greater than zero, review the messages carefully.

ADB2GEN condition codes

The ADB2GEN program is used to create a version file. The following condition codes are issued by ADB2GEN:

0
The program ran successfully.

4

- A parameter error occurred. The parameter was ignored or the default was used. No generate requests were issued.
- A 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 for z/OS system. This error is an internal error or limitation.
- Other severe errors were detected.

16
A severe error occurred.

GOC2CMP condition codes

The following condition codes are issued by the GOC2CMP program:

0
The program ran successfully.

4
The index was 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
Problems with referential constraints were detected. Manual action is required. Refer to the listed error message.

16
A severe error occurred. Refer to the listed error message.

GOC2DTC condition codes

In addition to the condition codes for GOC2CMP, GOC2DTC can issue the following condition code:

12
A quoted string is too long.

Troubleshooting: The Compare report shows changes to bind options for trigger packages

If the report from Object Comparison Tool includes unexpected changes to bind options for trigger packages, you might need to rebind some packages. This situation can occur when you migrate to Db2 11 or a later version.

Symptom: The compare report includes unexpected changes to bind options for trigger packages, as shown in the following example:

```
Compare Trigger source <table_schema>.<table_name> and target <table_schema>.<table_name>
Source type : <trigger_type> Target type : <trigger_type>
(A)Field SYSTEM_TIME SENSITIVE changed from YES to NO
(A)Field BUSINESS_TIME SENSITIVE changed from YES to NO
(A)Field ARCHIVE SENSITIVE changed from YES to NO
```

Explanation: When a trigger is created, the following fields in the SYSPACKAGE table have a default value of YES:

- SYSTIMESENSITIVE
- BUSTIMESENSITIVE
- ARCHIVESENSITIVE

These values are stored in packages at the time they are bound. Check these field attributes in your trigger packages. If you set them to NO on your old system and then these trigger packages are created on a new system, the default values of these fields on the new system are YES.

Solution: Rebind the packages.

To rebind the packages:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option I, and press Enter.
2. On the **DB2I PRIMARY OPTION MENU** panel, specify option 5, and press Enter.
3. On the **BIND/REBIND/FREE** panel, specify option 6, and press Enter.
4. Change the **PLAN MANAGEMENT** field to OFF.
5. Change the **SYSTEM_TIME SENSITIVE**, **BUSINESS_TIME SENSITIVE**, and **ARCHIVE SENSITIVE** field to the desired values.

Related information

BIND and REBIND options for packages, plans, and services (Db2 13 for z/OS)

Notices

This information was developed for products and services offered in the U.S.A.

This material may be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

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 Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119

Armonk, NY 10504-1785
US.

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 is for planning purposes only. The information herein is subject to change before the products described become available.

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.

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

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

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

IBM, the IBM logo, and ibm.com® are trademarks or registered marks 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.

Linux® is a registered trademark of Linus Torvalds in the United States, other countries, or both.

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

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java™ and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions:

Applicability: These terms and conditions are in addition to any terms of use for the IBM website.

Personal use: You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use: You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights: Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

Index

E

F

G

I

J

L

M

N

O

P

R

S

T

U

V

Product Number: 5655-CH1

SC28-2787-00